Aalto University

School of Science

Degree Programme of Computer Science and Engineering

Juho Sorvettula

Adopting agile software development in a small company – a case study

Master's Thesis

Espoo, March 20th, 2013

Supervisor: Professor Casper Lassenius, D.Sc. (tech)

Instructor: Jarno Vähäniitty, D.Sc. (tech)

Aalto-yliopisto Perustieteiden korkeakoulu Tietotekniikan koulutusohjelma		DIPLOMITYÖN TIIVISTELMÄ
Tekijä: Juho Sorvettula		
Työn nimi: Ketterän ohjelmistotuotannon käyttöönotto pienessä yrityksessä - tapaustutkimus		
Sivumäärä: 5 + 67	Päiväys: 20.3.2013	Julkaisukieli: Englanti
Professuuri: Ohjelmistotuotanto ja -liiketoiminta	Professuurikoodi: T3003	
Työn valvoja: Professori Casper Lassenius, TkT		
Työn ohjaaja: Jarno Vähäniitty, TkT		

Tiivistelmä:

Pienet ohjelmistoyritykset kohtaavat toiminnassaan usein haasteita, jotka liittyvät resursointiin ja priorisointiin. Etenkin ilman ulkopuolista rahoitusta toimivien yritysten tapauksessa riittävä kassavirta on turvattava asiakasprojektien avulla samalla, kun yrityksen olemassa olevia tuotteita on kehitettävä sekä uusia tuotava markkinoille. Tilanne näkyy usein kehittäjien työssä jatkuvina keskeytyksinä ja suunnitelmien muutoksina.

Muun muassa näiden haasteiden ratkomiseen suunnitellut ketterät ohjelmistotuotannon menetelmät ovat saavuttaneet suurta suosiota ohjelmistoyritysten keskuudessa huolimatta siitä, että ketterää ohjelmistokehitystä koskeva tutkimustieto on usein puutteellista ja ristiriitaista. Ongelmien syyksi on esitetty myös puutteellista tekemissalkun hallintaa.

Tässä työssä tutkittiin voidaanko tyypillisiä pienen ohjelmistoyrityksen ongelmia resursoinnissa ja tekemissalkun hallinnassa helpottaa ottamalla käyttöön ketterä ohjelmistotuotantomenetelmä. Työ toteutettiin tutkimalla 8-henkisen suomalaisen ohjelmistoyrityksen toimintaa yhdistetyn kysely- ja haastattelututkimuksen avulla. Tutkimus on toteutettu kahdesti: ensin ennen ketterän Scrum-prosessin käyttöönottoa ja toistamiseen yrityksen käytettyä Scrumia kolmen kuukauden ajan.

Sekä kyselyn että haastattelun perusteella havaittiin selkeää positiivista kehitystä yrityksen ohjelmistotuotannossa ketterän ohjelmistotuotannon käyttöönoton myötä. Tulos oli saman tyyppinen sekä kehittäjien että heidän esimiehiensä osalta. Suurin vaikutus ketterän menetelmän käyttöönotolla oli päivittäisen työn hallintaan. Tekemissalkun hallintaan menetelmä antoi valmiuksia, muttei suoria ratkaisuja.

Asiasanat: Scrum, ketterä ohjelmistokehitys, tekemissalkun hallinta, tapaustutkimus

Aalto University School of Science Degree Programme in computer science and engineering		ABSTRACT OF THE MASTER'S THESIS
Author: Juho Sorvettula		
Title: Adopting agile software development in a small company – a case study		
Number of pages: 5 + 67	Date: March 20th 2013	Language: Englilsh
Professorship: Software Engineering and Business		Code: T3003
Supervisor: Professor Casper Lassenius, D. Sc. (tech)		
Instructor: Jarno Vähäniitty, D. Sc. (tech)		

Abstract:

Small software companies often face challenges in resource allocation and prioritization. Especially companies operating without external funding must ensure adequate cash flow while constantly improving current products and bringing new ones to market. This situation reflects to developers' daily work as constant interruptions and changes in plans.

Agile software development methods, which were created to help solving these and other challenges, have received wide popularity amongst software companies even though the empirical evidence concerning agile methods is somewhat inadequate and conflicting. Inadequate development portfolio management has also been reported to cause these problems.

This thesis studies if typical small software company problems in resourcing and development portfolio managemen can be alleviated by adopting agile software development. The research data was collected by conducting a combined survey and interview in a Finnish sofware company with 8 employees. The study was repeated twice: first before adopting an agile process and second after using Scrum for three months.

Sekä kyselyn että haastattelun perusteella havaittiin selkeää positiivista kehitystä yrityksen ohjelmistotuotannossa ketterän ohjelmistotuotannon käyttöönoton myötä. Tulos oli saman tyyppinen sekä kehittäjien että heidän esimiehiensä osalta. Suurin vaikutus ketterän menetelmän käyttöönotolla oli päivittäisen työn hallintaan. Tekemissalkun hallintaan menetelmä antoi valmiuksia, muttei suoria ratkaisuja.

Both the survey and the interview indicated a clear positive trend in the case company's software development after adopting the agile process. The result was similar both for developers and managers. The agile process had the largest impact in managing the day-to-day work. The agile process was found to give tools for development portfolio management but to lack direct solutions.

Keywords: Scrum, agile, software development, portfolio management, case study

TABLE OF CONTENTS

1 INTRODUCTION

This section presents the motivation for this research. The background of the research field and the case company are presented, followed by the research problem. Also an overview of the structure of this thesis is given.

1.1 Background of the research

The IT industry is operating in a turbulent and constantly changing environment and the competition for new customers is fierce. At the Internet age when anybody is able to deliver software to all corners of the earth the software market is truly global. When users get products like Facebook or Gmail free of charge it puts pressure on the pricing of other types of systems as well, including business related software.

Historically the approach to building software has had an emphasis on planning (Boehm 2002). The problems that were to be solved by the help of software were carefully analyzed and comprehensive requirements documentation was produced before writing the actual program code. A lot of the resources used for the software development were wasted in extensive project management tasks.

Today the practitioners have widely adopted so called agile software development methods (West & Grant 2010), such as Scrum, that aim to solve the complex problem of building software by focusing on controlling change rather than fixing plans beforehand. Although the adoption of these methods is considered to be substantial among the industry the scientific information available about them is somewhat conflicting and incomplete (Dybå & Dingsøyr 2008).

1.2 The case company

The case company is a Finnish software company focusing on automating social service and health care customer processes. The product range consists of two main products that are both offered to the customers as web applications. All customers use the same production instance of the software, so no client-specific installations exist. This means that instead of selling licences the business model is to offer the products as software-as-a-service. The customer pays a monthly fee for the purchased products with some transaction fees added when necessary.

Despite this the system is not identical for all clients. A major part of of the case company's business is process improvement consulting for health care facilities. This often includes customizing the feature set in their products for individual clients or adding custom features. Using the product also changes the way certain processes are taken care of for example in health care organizations. This means that when the product or products

are introduced in a new organization, training the staff and making sure additional instructions are available is very important.

The company is privately owned and has no extensive outside funding. This means that all development efforts for future projects and products are funded by the sales of the existing services.

The development personnel in the case company are using their time for the following types of activities:

- Release-based development
- Maintenance of existing products
- Client-specific development
- Consulting
- Deliveries & training

Additionally, these activities must be performed on two separate products.

The staff consists of the following personnel:

- Four-member software development team
- A person responsible for user service and guidance
- Two sales representatives
- CEO

1.3 Research problem

Based on the above it can be stated that the case company is operating in a multi-product and multi-project environment where different activities compete for the same development resources. Some of the biggest challenges include deciding how to distribute the limited resources between competing development activities and making sure these decisions are actually followed trough.

From the development team's point of view these challenges are often manifested as interruptions and sudden changes in the plan-of-action. The developers often have to discontinue the development task at hand in order to address a problem elsewhere. This creates a stressfull work environment that often feels overly hectic.

In their own opinion the developers and managers feel that the problem is not necessarily just the fact that the development team is small and they have a lot to do. It seems likely that with better methods of organizing work, communicating the plans and possible changes to them more efficiently and by having less distractions from the task at hand the same development team could accomplish more.

The deliverable of this project is to improve the software development process in the case company so that these problems are addressed. The main objective for the new process is

to enable the stakeholders to make informed decisions about resource alignment between different activities. The main objective is supported by the following sub-objectives:

- The developers should be able to focus on their work without sudden interruptions
- The process should enable changing plans in a controlled manner
- The stakeholders should be able to follow how their decisions are reflected into the daily work of developers

Based on these objectives the research problem of this study is:

> *Can adopting an agile software process alleviate the problems recognized in the daily development activities of the case company?*

The research problem is divided into 5 research questions.

1. *Why can it be assumed that improving development portfolio management could help solving the problems?*
2. *Why can it be assumed that using an agile software development process could help solving the problems?*
3. *Does the ability of the different stakeholders to make informed decisions about resource alignment improve after implementing the process model?*
4. *Does the ability of the development team to focus on current work improve after implementing the process model?*
5. *Does the ability of the stakeholders to follow how their decisions are reflected into the daily work of developers improve after implementing the process model?*

1.4 Structure of the thesis

The remainder of this thesis is organized as follows. Section 2 presents the concepts of development portfolio management and agile software development and essential background information on Scrum and its usage in environments similar to the case company are presented. Section 2 aims to find an answer for the two first research questions. Section 3 presents the study methodology used in this thesis. Sections 4, 5 and 6 discuss the implementation of the new software development process in the case company and present the findings from the study. The answers for research questions 3,4 and 5 are based on these sections. In section 7 the validity and future implications of the results are discussed and an overview of the conclusions and findings from the case company is given.

2 LITERATURE STUDY

This section aims to answer the first two research questions presented in section 1.3. The first subsection explains the concept of development portfolio management in the context of this thesis and seeks to justify its importance for the case company. The second subsection presents agile software develoment methods and their connection with the so-called traditional methods and lists some examples. Also critical success factors and challenges of agile adoption are discussed. The third subsection presents the Scrum process model.

2.1 Portfolio management

Portfolio management is a concept used in new product development that refers to the process of allocating company's research and development resources by evaluating and prioritizing new and existing projects. The aim is to optimize the portfolio in terms of value of the portfolio, strategic alignment, balance of risk and number of projects (Cooper et al. 2002).

In the context of this thesis portfolio management refers to the process of managing the product development portfolio of a software company practising new product development. Vähäniitty (2010) defines this kind of portfolio management as:

> *"Portfolio management refers to the decision-making process for updating and revising a business's product development portfolio, that is, the list of active and planned development activities that require the development resources' attention."*

Portfolio management is often considered an activity only needed in large corporations, but as Vähäniitty et al. (2010) point out, also small software companies like the case company of this study can benefit significantly from well conducted portfolio management. It is especially important for small companies that have to balance between maintaining cashflow trough customer projects and driving their own product development forward at the same time with scarce resources, which is the situation faced in the case company.

Vähäniitty et al. (2010) also present a list of eight typical problems that have been associated with inadequate or inefficient portfolio management in literature:

1. Excessive multitasking
2. Firefighting
3. Overload
4. Ineffective decision-making
5. Missing strategic alignment

6. Slipping schedules
7. Project failures and poor profitability
8. Perceived need to improve project management

These problems seem to be well linked with the problems identified in the case company that were listed in section 1.3. The frequent changes of plans, inability to focus on a single task at a time and constant busyness at work that are experienced in the case company map to problems 4, 1 and 3 from the typical problem list presented above. The employees in the case company have also recognized the need to improve project management.

Thus, four out of eight typical symptoms arising from inadequate portfolio management can be identified at the case company. This suggests that by improving the portfolio management the problems experienced in the everyday development work could be alleviated.

The concept of portfolio management is not new in new product development literature (Cooper et al. 2001) and numerous methods for achieving sufficient level of portfolio management have been suggested (Cooper et al. 2001, McGrath 1996, Wheelwright & Clark 1992). However, these methods typically focus on large company perspective and are only compatible with traditional software development methods. The more modern approaches suggested for example by Rothman (2009) and Poppendieck & Poppendieck (2010) aim to combine expicit portfolio management with so-called agile software development methods. Rothman even suggests that portfolio management would benefit from using agile methods.

2.2 Agile software development

Agile software development is an umbrella term that consists of multiple practices, methods and process models. This subsection presents the origin and basic ideas behind the software development methods that are considered agile. Where as the previous section presented the concept of portfolio management and underlined the importance of it also for small companies like the case company this section aims to explain how an adequate level of it could be achieved by using the agile methods.

2.2.1 Origins of agile software development methods

The most common example of a traditional software development method mentioned in the end of section 2.1 is the waterfall model presented by Royce (1970). The model is based on the idea of dividing software development work into subactivities that conribute to the final product. According to Royce the minimum of steps needed to implement a computer program are analysis and programming. As this approach only suites very small-scale projects he suggests a more comprehensive set of subactivities to be used for larger applications. Typically these steps include:

- Requirements specification
- Analysis
- Design
- Coding
- Testing
- Maintenance

The waterfall model is often comprehended as sequential: each subactivity must be finished before starting work on the next. This interpretation is somewhat conflicting, since Royce refers to the iterative nature of the software development work multiple times in his original article.

Regardless of Royce's intentions, the prevailing way of building software with the waterfall model is sequential and involves phase-gate type of project management (Larman 2004). This means that for example after the analysis phase there is a formal evaluation session where all the produced analysis documentation is assessed and only after official approval the design work can be started. The traditional approach also assumes that it is possible to fully specify the problem field before starting to create the software, that an optimal solution for the problem exists and that it is possible to create the finished software with a singe pass trough all of the steps of the waterfall (Dybå & Dingsøyr 2008).

The agile development methods can be seen as a counter-reaction to these traditional software development methods, which are viewed as inflexible and bureaucratic. The attempt to force sequentiality into the problem solving process that is iterative by nature is among the key problems, which agile methods try to solve. The agile movement claims that the assumption that the software and its requirements can be fully specified before implementing it is wrong.

This is why focus in agile methods is shifted into delivering working software early, trusting skilled individuals and being able to control rapid change (Jiang & Eberlein 2009). By delivering working software to end-users in early stages of the project the developers are able to get feedback and make changes in the plans if needed.

The origins of the agile movement can be traced back to the lean production philosophy (Poppendieck 2005) developed at large Japanese enterprises such as Toyota (Liker 2004) and Honda (Holford and Ebrahimi 2007). The agile approach to software development is described in the Agile Manifesto (Beck et al. 2001), which states the following:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*

- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

That is, while there is value in the items on the right, we value the items on the left more."

One of the main methods for accomplishing the aforementioned qualities in a software development process is to use time pacing in modifying the process from sequential to iterative. Instead of first describing all the features of the end product the agile movement suggests to just specify a few, implement them and ask the users how they like it. After this, another set of features can be specified, designed, implemented and tested. These cycles, often called iterations or sprints, form the heart of many of the agile methods.

Several software development methods have been presented that are considered agile. A comprehensive analysis on different agile methods is out of the scope of this research, but in order to give an overview of the field a description of six methods are presented in table 1 on the next page. The table includes six "main agile development methods" identified by Dybå and Dingsøyr (2008).

When comparing agile methods it must be noted that they are very different in nature. For example Scrum is a process model, which does not dictate how to arrange the practical work of producing program code where as XP focuses more on useful coding practices and less on the process side of things. It is not uncommon to combine Scrum with some of the XP practices for example by using pair programming or continuous integration in conjunction with Scrum (Mar and Schwaber 2002).

However, there are also studies that challenge the separation between "traditional" and "agile" methods. Jiang and Eberlein (2009) analyzed and compared the history of both types of methodologies and found that they share common roots and could be seen as complementary. They argue that there is no single solution that covers all types of software projects and environments and that the methods and process models used in software engineering should be carefully selected for each project. It should also be noted, that the original article on the waterfall model by Royce actually describes the process as iterative, and that the subactivities he describes (requirements specification, analysis, design, coding, testing and maintenance) are still being used in the agile methods.

Table 1: Comparison of main agile development methods

Method	Description	References
Crystal methodologies	The Crystal methodologies are a family of methods that acknowledge the need for different methods in different kinds of projects. The methods include Clear, Yellow, Orange, Red, Blue, where the color changes depending on the criticality and size of the project. The most agile method is the Crystal Clear. The Crystal methodologies do not limit development practices so the use of XP or Scrum practices as part of the process is possible. Crystal Clear method has seven main features that describe the process: frequent delivery, reflective improvement, osmotic communication, personal safety, focus, easy access to expert users, and requirements for the technical environment. It requires the development team to be co-located.	Cockburn 2004
Dynamic systems development method	The current version of DSDM is called Atern and it is intended for effective project management and delivery. DSDM also aims to be tool and technique independet. DSDM Divides projects in four phases: study, functional model iteration, design and build iteration and finally implementation. Eight principles underlie DSDM: focus on business need, delivering on time, collaboration, never compromising quality, building incrementally from firm foundation, developing iteratively, demonstrating control and continuous and clear communication	Stapleton 2003, Craddock et al. 2008
Feature-driven development	A combination of model-driven and agile approaches that focuses on translating the user needs into working features. Divides work into five main activities: developing overall model, building feature list, planning by feature, designing by feature and building by feature.	Palmer & Felsing 2002
Lean software development	A translation of lean production and, in particular, the Toyota production system to software development. Can be summarized by seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole.	Poppendieck & Poppendieck 2003
Scrum	Scrum is based on the idea of incremental and iterative development and is characterized by time pacing. In Scrum the software is developed in small increments during relatively short iterations and the emphasis is on delivering working features for the client very early in the project. The work is ordered into proritized lists called backlogs. Currently the most adopted process model used in agile software development.	Schwaber 1995, Schwaber & Beedle 2001, Schwaber & Sutherland 2011
Extreme programming (XP, XP2)	Main focus on the best practices for development. The advocated practices include: pair programming, code reviews, automated testing, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers.	Beck 2004

2.2.2 Benefits of agile software development

Abrahamsson et al. (2002) were among the first to review the existing literature on agile software development methods and they found that a major part of the research was either non-empirical or lessons-learned type of studies. They found anecdotal evidence about the suitability and effectiveness of agile methods for many situations but stress the fact that the actual empirical evidence backing this claim is scarce.

Dybå and Dingsøyr (2008) provide a more recent and more systematic review to the existing empirical research on agile methodologies. After assesing 1 996 articles and reducing them to 36 most relevant they found results that suggest the following benefits for using agile methods:

- Agile methods are often easy to adopt and work well in different kinds of environments
- Customer collaboration and customer satisfaction is increased
- Ability to focus on current work is increased
- Estimation is improved
- Process of handling defects is improved
- Some comparative studies found improvements in productivity of developers
- Improvements in quality of software

However, Dybå and Dingsøyr also found that the strength of evidence backing these claims is very low and that it is difficult to make estimations about the benefits of agile methods to a certain organization or project before actually applying them.

Despite the fact that scientific proof about the superiority of agile methods compared to the traditional ones is at best very limited the practitioners seem to rely heavily on them. In a questionnaire by West and Grant (2010) 35% of the respondents reported using agile methods, and the proportion can be expected to have risen since. It is possible that since the concept of agile software development is relatively new and since software development is inherently complex, it takes time to build the body of knowledge enough for it to explain the popularity of agile methods.

2.2.3 Adopting agile process models

Niazi et al. (2003) found that the literature concerning software process improvement (SPI) has concentrated mostly on describing the preferred end results of SPI initiatives but less research exists on how to implement the improvements. Case studies that describe the agile transition of an organization do exist, but often revolve around large organizations rather than small companies like the case company of this study (Pikkarainen et al. 2005, Benefield 2008). These studies suggest a step-by-step approach, where the transition starts with a pilot phase and is done one team at a time.

Börjesson and Mathiassen (2004) state that in order to succeed SPI initiatives need 4 things:

- Stakeholder commitment
- Organizational learning
- Distributed resources
- Customer relations

They also advise to expect chaos in the beginning and highlight the importance of developing the new process by iterating.

Dybå (2005) and Chow & Cao (2008) have studied the critical success factors in agile software projects. The critical success factors identified by Chow and Cao are correct software delivery strategy, a proper practice of agile software engineering techniques and a high-caliber team. Three other factors that could be critical to certain success dimensions were found to be a good agile project management process, an agile-friendly team environment, and a strong customer involvement. Dybå suggests the following success factors: correct alignment between SPI and business goals, employee participation, involved leadership, concern for measurement, exploitation of existing knowledge and exploration of new knowledge.

The factors identified by the studies differ in their wording, but similarities can be found. All three studies stress the importance of the people factor, i.e. the willingness of both the development team and the management to make an effort to improve the process and to learn new ways of working. Also the importance of customer involvement and alignment with business goals are mentioned in at least two of the studies.

In practice it seems that the favoured method for implementing an agile process such as Scrum in a small organization is quite straightforward. Agile methods often include built-in methods and events for assessing the process in a retrospective way and these methods make it possible to start using agile processes without extensive preparation. If problems arise they can be tackled immediately.

2.3 The Scrum process model

The Scrum process model is an agile software development process framework first suggessted by Schwaber (1995). Scrum is based on the idea of incremental and iterative development and is characterized by time pacing. In Scrum the software is developed in small increments during relatively short iterations and the emphasis is on delivering working features for the client very early in the project. Scrum is currently the most adopted process model used in agile software development (West & Grant 2010).

The explanation of Scrum process given in the following subchapters is based on the Scrum guide maintained by Schwaber and Sutherland (2011) unless otherwise stated.

2.3.1 Roles in Scrum

Scrum team is a combination of the development team, a Scrum master and a product owner. The development team consists of the actual development workforce. The ideal development team for Scrum is self-organizing and crossfunctional so that they are able to choose their working methods independently and solve problems without external help.

The Scrum master is a role that roughly translates into a 'project manager' or 'team leader' in the more traditional types of organizations (Schwaber 2004). Scrum master in responsible for enforcing the process and removing obstacles that prevent the team from accomplishing their goals. The Scrum master also 'protects' the team from unnecessary external disturbance during the sprint. Scrum master is also responsible for making other stakeholders and members of the organization understand the process and understand how they can make their actions more beneficial for it.

The product owner is the owner of the problem the team is trying to solve. The product owner is responsible for maintaining and prioritizing the backlog and delivering the necessary requirements for the team.

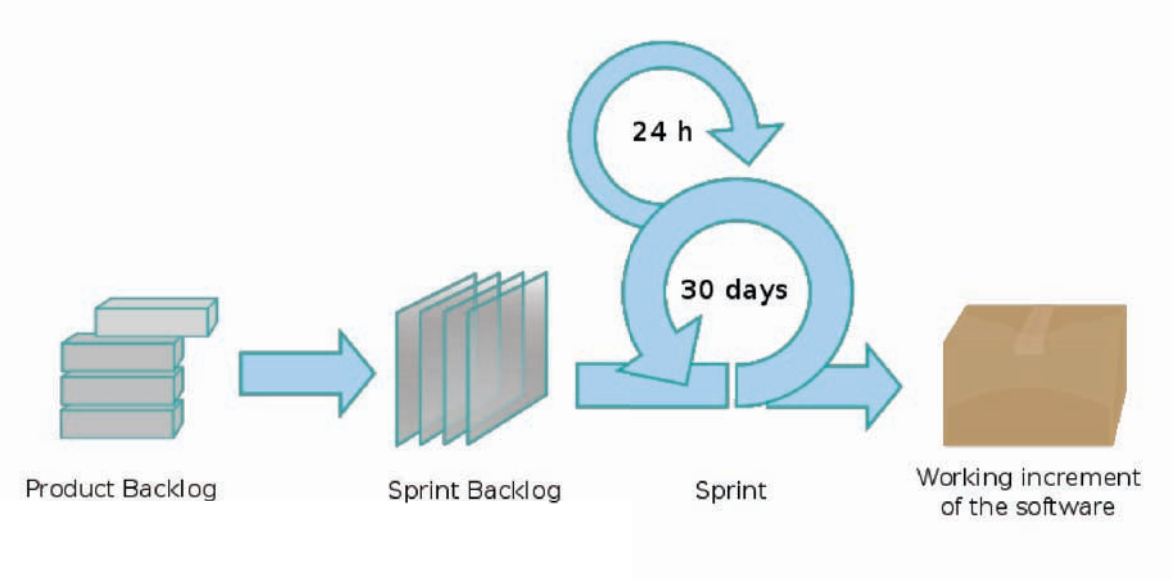


Figure 1: Scrum process model in a nutshell (Wikimedia Commons, 2008)

2.3.2 Scrum events

Scrum sprint, often also called iteration, is typically 2 – 4 weeks long in duration. The sprint is a predetermined timeframe during which the actual development work is done. During the sprint the team members complete their development work and participate in a meeting called daily scrum every day. The daily scrum meeting is a short status check that only lasts about 10 minutes. During the daily scrum meeting every team member answers three short questions:

1. What have I done since the last daily scrum meeting?
2. What shall I be doing before the next scrum meeting?
3. What problems have I encountered?

Figure 1 presents an overview of the scrum sprint and the daily scrums: The vision of the product owner is artivulated in the backlog items that are prioritized in the product backlog. The items on the top of the priority list are selected into the sprint backlog and the development team completes them during the sprint and monitors the situation with the help of daily scrum meetings. After the iteration a new, possibly shippable increment of the software exists.

The sprint begins with a sprint planning session that is typically divided in two parts: a rough estimation of the contents of the sprint followed by a more detailed analysis on how to complete the plan. The first part of the sprint planning aims to answer the question "what is done during the sprint?" and the second part the question "how is the work done?". The purpose of this two-fold strategy is to first get a quick overview of the situation and find out which of the planned work items could possibly be completed during the sprint. Only these pre-qualified work items are then inspected in detail in order to save time by not having to estimate the effort for every possible backlog item.

A typical way of making the preliminary estimation is to use a combination of two methods called planning poker and story points. A story point is an arbitrary unit of work that describes the size and complexity of a user story. It could be compared to estimating effort in man-days or person working weeks. Story points are usually estimated on a fixed scale, for example a Fibonacci series (1-2-3-5-8-13-…), and the amount of points are selected by playing planning poker.

In planning poker all developers have a set of playing cards with the different point amounts in their hands. A backlog item is presented to them and all developers select the card that represents the amount of story points it takes to complete the backlog item in their opinion. Once everyone is ready the selected cards are revealed to others. If everybody agrees on an amount the issue is settled. If not, the developers discuss the topic and try a new voting until a consensus is found.

The size of story points varies greatly across different organizations, in some cases a story point can match roughly to just an hour of development work and in others it can mean a whole work day. This doesn't matter as long as the size of the point stays relatively stable inside the organization. With the combination of story point estimation and planning poker large amounts of backlog items can be quickly evaluated and ranked.

The sprint ends in a wrap-up session consisting of a demonstration of the completed work and a reflection meeting often referred to as a retrospective. During the demonstration the team presents the work it has completed during the sprint to the product owner and other

stakeholders. This is very important in order to make sure that the completed work meets the requirements and that all stakeholders understand how it changes the product.

The demonstration focuses only on the results delivered during the iteration. The process is assessed in the retrospective. Usually the retrospective is a short meeting where the development team gathers a list of things that went well and another list that includes possible problems. The goal is also to find suggestions for solutions that can be tried in the next iteration. These suggestions are again evaluated in the following retrospective and possibly developed further.

2.3.3 Scrum artifacts

The most important artifacts in Scrum are the backlogs together with the backlog items they contain. A backlog is a prioritized list of work items. Scrum splits the backlog in two parts: the product backlog and the sprint backlog.

A product backlog includes all backlog items of a product. The level of detail is relatively low and the amount of work needed to implement product backlog items can vary greatly. Once the backlog items have a high enough priority they are moved into the sprint backlog where they get implemented. At this point it may be necessary to split large backlog items into smaller parts in order to make them fit in a sprint and often also additional information is needed.

The backlog items are often presented in the form of user stories. User stories are a lightweight and nimbe alternative for the traditional format of software requirements documentation. Leffingwell and Behrens (2009) define a user story as:

> *"A User Story is a brief statement of intent that describes something the system needs to do for the user."*

The user story has an emphasis on the value the feature delivers for the user. A basic example of a user story for a web application might be:

> *"As a frequent user I want the system to remember my login on my own computer so that I don't have to type my username and password every time I visit the application."*

Before implementation the user story is supplemented with discussions between the Product Owner and the developers at the latest during the iteration planning sessions. Also more detailed specification documentation can be attached to the user story.

Todether the user story, the discussions between stakeholders and the attached requirements documentation form the backlog item that can be prioritized and assessed.

The situation of the ongoing sprint is usually visualized as a burdown chart that shows the amount of effort left for the team to complete the sprint. The chart typically has a reference

line depicting the ideal constant-velocity line and the actual line for the effort left updated daily.

The burndown can be drawn to a whiteboard or in digital format, but the central requirement is that it must be visible to all developers, the scrum master and the product owner during the iteration in order to maintain a common understanding about the ongoing situation.

2.3.4 Definition of done

Definition of done is an organization specific, written listing of the criteria that must be fulfilled in order to be able to call a backlog item "done". It typically includes things like "all acceptance tests pass", "code review with a senior developer is conducted" or "backlog item is reported ready in a daily scrum meeting". The definition of done evolves with the rest of the process and as the team and organization become more mature as Scrum users so does the definition of done.

2.3.5 Tools for Scrum

The case company wanted a web based software application to act as the central tool for managing the different backlogs and keeping track of the iterations. The requirements for the backlog management tool included:

- Preferrably free-and-open-source
- Active development community
- Ability to host it on company servers
- Good support for working away from the office
- Includes portfolio management features

Sourceforge website (www.sourceforge.net) was searched to find open source projects that would meet these requirements. Many possible alternatives were found but the only tool that included portfolio management features was Agilefant.

Agilefant (www.agilefant.org) is an open-source backlog management tool that aims to be simple yet scalable. In addition to the basic features needed for Scrum, which include the backlogs and burndowns, it also includes project- and portfolio-level planning tools.

Agilefant has a three-layer backlog where the product sits at the bottom layer. The product backlog includes all stories related to the product, both already completed and future work. The next layer is called a release project and it has its own backlog. The iterations belong to a release project and represent a subset of the release projects backlog. In addition to the backlog views Agilefant also has a view called Daily Work that shows all the work currently assigned to a single developer. An example of the daily work view is presented in figure 2 and an example of the story tree for a product is viewed in figure 3.

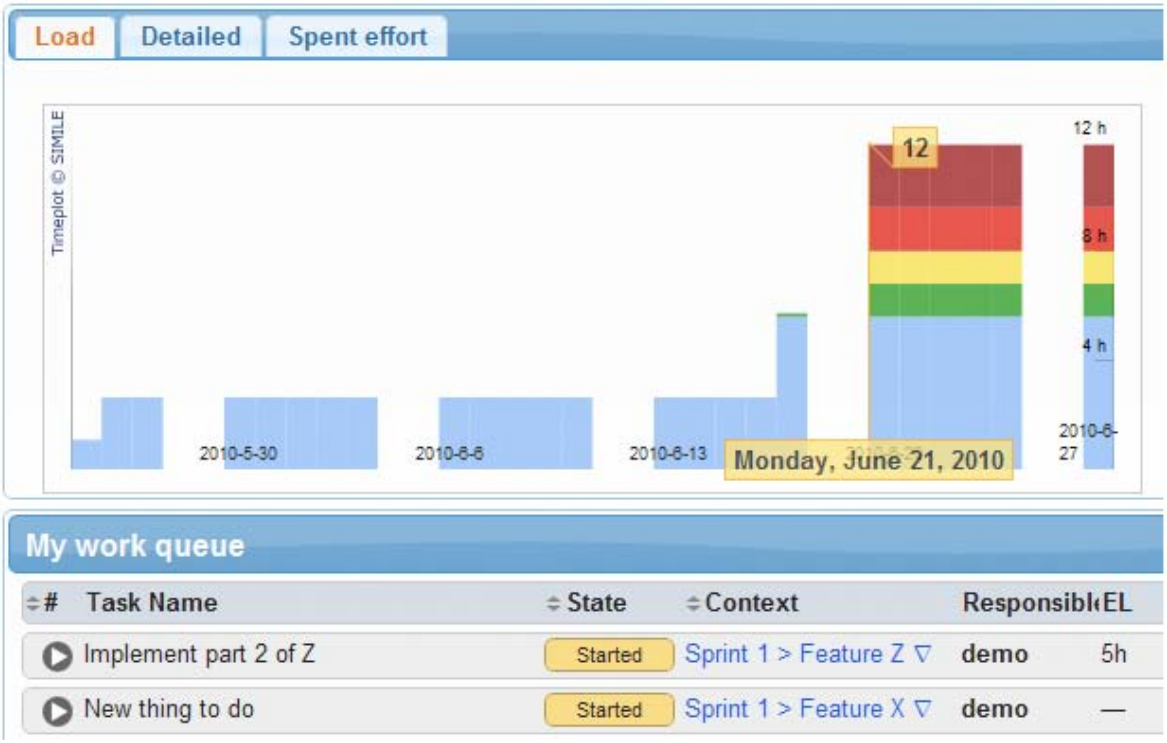


Figure 2: Daily work view in Agilefant software

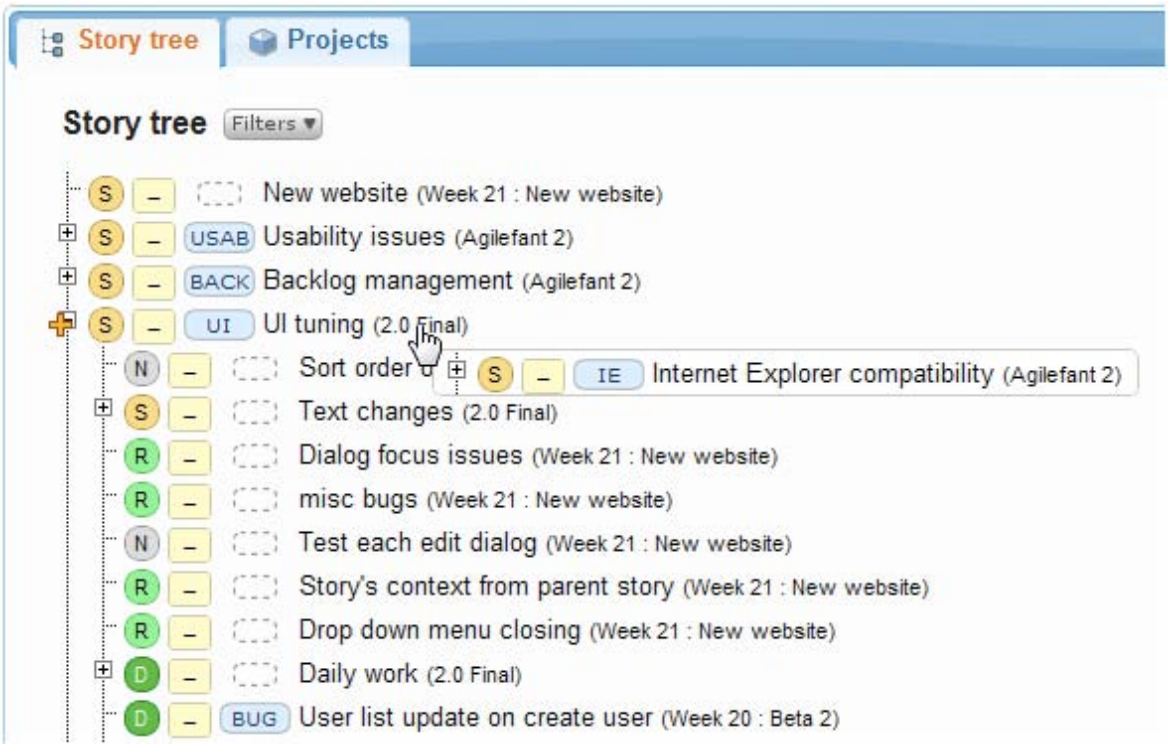


Figure 3: Product story tree view in Agilefant software

3 STUDY METHODOLOGY

According to Humphrey (1989) the traditional scientific approach to evaluating process improvement success is to compare quality and productivity metrics between the old and new practices. The problem with this basic starting point in this study was that the case company was not collecting any systematic quality or productivity data before the process improvements described in this thesis started.

The solution was to conduct an assessment of the current situation and repeat the same evaluation after implementing the new process. The assessment was conducted by using the development portfolio health barometer –method presented by Heikkilä and Rautiainen (2010), which is a combination of a survey and a semi-structured interview.

The development portfolio health barometer is a systematc and structured method for analyzing the case organization's software development portfolio management process and its possible problems. It is based on a systematic literature review by Vähäniitty (2006), which studied what problems in software companies are connected to inadequate portfolio management. As explained in section 2.1 the problems faced by the case company are quite typical portfolio management problems and since the health barometer measures the same issues it suites well for this study.

This chapter presents the survey part of the health barometer in the first subsection and the interview part in the second. Both conducting the research and analyzing the data are discussed. The third subsection presents the detailed research setup for this thesis.

3.1 Survey

The survey part of the development portfolio health barometer consists of 52 statements that assess the health of the development portfolio management practices and principles of the company. The statements are presented in tables 2, 3 and 4. Additional demographic information can also be collected. The assessment is divided into three main categories, which are hereditary factors, lifestyle issues and symptoms.

The hereditary factors describe the environment of the case organization. They refer to properties that are in fact outside of the scope of development portfolio management but have an impact on the need for it. Unlike lifestyle issues and symptoms, the results from the statements in hereditary factors are not necessarily 'good' or 'bad' as such. They simply describe the situation the studied organization works in. However, these scores indicate the amount and rigor needed for adequate development portfolio management. Hereditary factors are not easy to change but this can also be done if enough effort and time are given. The statements used to measure hereditary factors are presented in table 2.

Table 2: Statements used to measure hereditary factors

Hereditary factors		
Issue		Statement
1	Leveraging customer-specific activities for product development	New products or features are developed in customer-specific projects
2	Multiple roles and responsibilities	Most of our development people have a broad work profile (e.g. they participate in many of the following: product development, customer projects, project management, sales / sales support, customer support, consulting, deliveries, training, etc.)
3	Dependency on cash flow	A downswing in cash flow is quickly reflected in the ability to pay salaries
4	Clarity of strategy: Definition	Strategy and long-term plans have been clearly defined
5	Clarity of strategy: Communication	Strategy and long-term plans have been clearly communicated
6	Appropriateness of incentive systems	Developers, project managers, sales, or senior managers are evaluated and rewarded in ways that are harmful to the whole
7	Appropriateness of organisational structure	Our organisational structure supports our current operations
8	Health of individual activities' practices	Each of our different activity types (e.g. product development projects, customer-specific development, maintenance, deliveries, etc.) has its own practices, that work

Table 3: Statements used to measure lifestyle issues

Lifestyle issues		
Issue		Statement
9	Identification of development activity types	We have identified the different types of activities development people spend their time on (e.g. product development projects, customer-specific development, maintenance, deliveries, etc.)
10	Managers' ability to see the 'big picture' (the development portfolio)	Business people are able to see the 'big picture' of ongoing activities (a.k.a. the development portfolio)
11	Developers' ability to see the 'big picture' (the development portfolio)	Development people are able to see the 'big picture' of ongoing activities (a.k.a. the development portfolio)
12	Target spending levels	I understand how much time, from a business perspective, I should spend on different types of activities
13	Criteria for selecting and prioritizing activities	We have criteria for prioritising our ongoing development activities
14	Prioritization of the portfolio	I understand the priorities between ongoing activities (e.g. project X vs. project Y, project X vs. support request Z, etc.)
15	Understanding of dependencies	I understand the dependencies of the ongoing activities
16	Managing the development activities as an explicit portfolio	All the ongoing and immediately upcoming activities that require attention from the developers are managed as an explicit portfolio
17	Clarity of role and responsibility definitions	We have defined who are responsible for development portfolio management
18	Clarity of roles and responsibilities in practice	It is clear who should participate in development-related decision making in different situations (e.g. in the middle of a project, when an urgent maintenance request arrives, when making an offer, when deploying a product, etc.)
19	Reflecting the portfolio to the company's strategy	We actively reflect the content of the development portfolio to the company's strategy
20	Considering the big picture in decision making	In decision making we mainly consider individual activities and do not take the "big picture" into account

Table 4: Statements used to measure symptoms

Symptoms		
Issue		Statement
21	Number of ongoing activities	How many different activities (product development projects, customer projects, etc.) are currently ongoing in your company?
22	Number of activities you are involved in	How many different activities (product development projects, customer projects, etc.) are currently ongoing that you are involved in?
23	Number of your own responsibilities	In addition to my main responsibility, I also have other, time-demanding responsibilities
24	Compromized throughput due to optimized resource utilization	We have too many parallel ongoing activities
25	Amount of parallel work in general	A single person is usually assigned to only one activity (e.g. a project) at the same time
26	Intentional limiting of work-in-progress	We complete one thing at a time and don't shift our attention from one incomplete task to another
27	Resource allocation by fire fighting	Fire fighting describes our work in practice
28	Ignoring resource allocations	Resources are being shifted from one activity (e.g. a project) to another regardless of previously agreed assignments
29	Cascading effect of resourcing changes	Changes in resourcing for one activity (e.g. a project) cause uncontrolled changes in other activities
30	Flexibility of resourcing	Resource commitments are too rigid for leveraging suddenly emerging opportunities
31	Working overtime	I work overtime
32	Pipeline pushing	When planning product releases or making offers, we consider how to resource the work in practice
33	Launch frequency of new activities	New activities (e.g. projects) are launched too often
34	Impact of busyness to work quality	Our employees have too much to do and quality of work suffers from it
35	Sufficiency of resources	We have enough resources in proportion to the amount of work
36	Pruning of non-essential activities	Activities (e.g. projects) are never killed
37	Management involvement in decisions regarding activities	If time runs out, developers resolve by themselves what can be left undone
38	Monitoring progress of activities	The real status of activities is known in development portfolio -level decision making
39	Rate of change of priorities	The priority ranking of activities changes constantly
40	Management response to problems	Management reacts to problems detected in activities (e.g. projects) too late
41	Senior mgmt's involvement in portfolio level decision-making	Senior management is actively involved in portfolio-level decision making
42	Health of the dialog between Business and Development	The dialogue between Business and Development people works
43	Strategic alignment of ongoing activities	Ongoing activities are in alignment with the company's strategy
44	Significance of ongoing activities	Ongoing activities are essential to our business
45	Portfolio balance: leveraging existing products	We have a sufficient amount of development projects that incrementally improve existing products or services
46	Portfolio balance: creating new business	We have a sufficient amount of product or service development projects that aim for new business
47	Progress of activities	Ongoing activities are behind schedule
48	Activity progress status reporting	Progress of ongoing activities is reported optimistically
49	Performance of the development organization	From a business viewpoint, development performs its duties well
50	Improvement in software development capability	Our capability to produce high-quality software has improved during the past year
51	Investing in improving practices of individual activities	We should invest more in improving the practices of individual activities (e.g. project mgmt., team practices, deployment processes, sales processes, customer support, etc.)
52	Investing in improving development portfolio management	We should invest more in improving development portfolio management (e.g. prioritizing activities, linking strategy with daily work, structuring the development portfolio, etc.)

Lifestyle issues measure the development portfolio management process of the company, the structure of the portfolio and the health of the practices used in the management process. The structure of the development portfolio refers to identifying and arranging the different ongoing and planned development activities in the organization. The statements for lifestyle issues are presented in table 3.

Symptoms refer to the issues arising from inadequate or ineffective development portfolio management. Symptoms are often easier to indentify and manage than hereditary factors or lifestyle issues, but focusing only on the symptoms does not solve the underlying problems. However, assessing the symptoms helps to identify the hereditary and lifestyle issues causing the problems. Statements regarding the symptoms are presented in table 4.

In the survey the subject answers each statement on a Likert-like scale ranging from 1 to 6. The end points of the scale are presented to the subject as “Strongly agree” (translates to 1 in the numeric scale) and “Strongly disagree” (translates to 6). There is also a “Don’t know” option available, which is handled as a blank answer. Furthermore, the subjects are able to comment on each answer but this is optional.

3.2 Semi-structured interview

The interview part of the development portfolio health barometer is based on the same statements as the survey. All of the subjects survey answers are discussed in order to gain insight into the reasons behind the numerical answers.

The basic workflow of the interview was that the interviewer and the subject had the subjects survey answers at hand and the subject was asked to briefly explain the rationale behind each answer. Depending on the ambiguousity of the answer some of the statements were discussed more in detail than others.

In order to further probe into the answers a technique called laddering was used during the interviews as suggested by Heikkilä and Rautiainen (2010). In laddering the questions “how” and “why” are used to control the level of concreteness or abstractness of the answers. After the initial answer the question “how” can be used to get more concrete examples thus reducing the probability of misunderstandings between the interviewer and the subject. By using the question “why” a more abstract level can be reached. This can be useful for gaining insight into the “big picture” behind the answers.

All statements of the survey are discussed during the first round with all of the subjects in order to properly understand the starting point of the case organization and the different stakeholders. On subsequent rounds the interview is more targeted towards the answers that have changed between rounds. This approach aims to reveal the effectiveness on ineffectiveness of different changes that hace been implemented between the health

barometer rounds. To ensure the validity of the results the answers from the previous round are shown to the subject during the time they give the answers for the current round.

The interviews were recorded with an internal microphone of a laptop computer in order to avoid disturbing the subject by making notes during the actual interview. The voice recordings were later transcribed into an excel spreadsheet. The actual spreadsheets are not included in the appendix of this study since the subjects felt that it would be too easy to recognize who gave which answers, but sections 4 and 6 include many quotes from the interviews.

3.3 Research setting

The research setting for this study had three steps:

1. Measurement of the current situation by conducting a development portfolio healt barometer round
2. Deployment of Scrum process in the case organization and using it for a period of time
3. Repeating the same development portfolio healt barometer measurement and analyzing the difference in results between rounds

The first health barometer round was conducted between 2nd and 11th November 2011 and the corresponding interviews were done between 10th and 18th November. Since the staff of the case company consists of only eight employees it was justified to collect survey answers from everyone. All of the developers and managers were interviewed so total amount of interviewees was seven. It was decided not to interview the person responsible for user service and guidance since her work duties do not include software development tasks. In an organization with eight members seven interviewees was considered enough to get a clear picture of the reasons behind the survey answers.

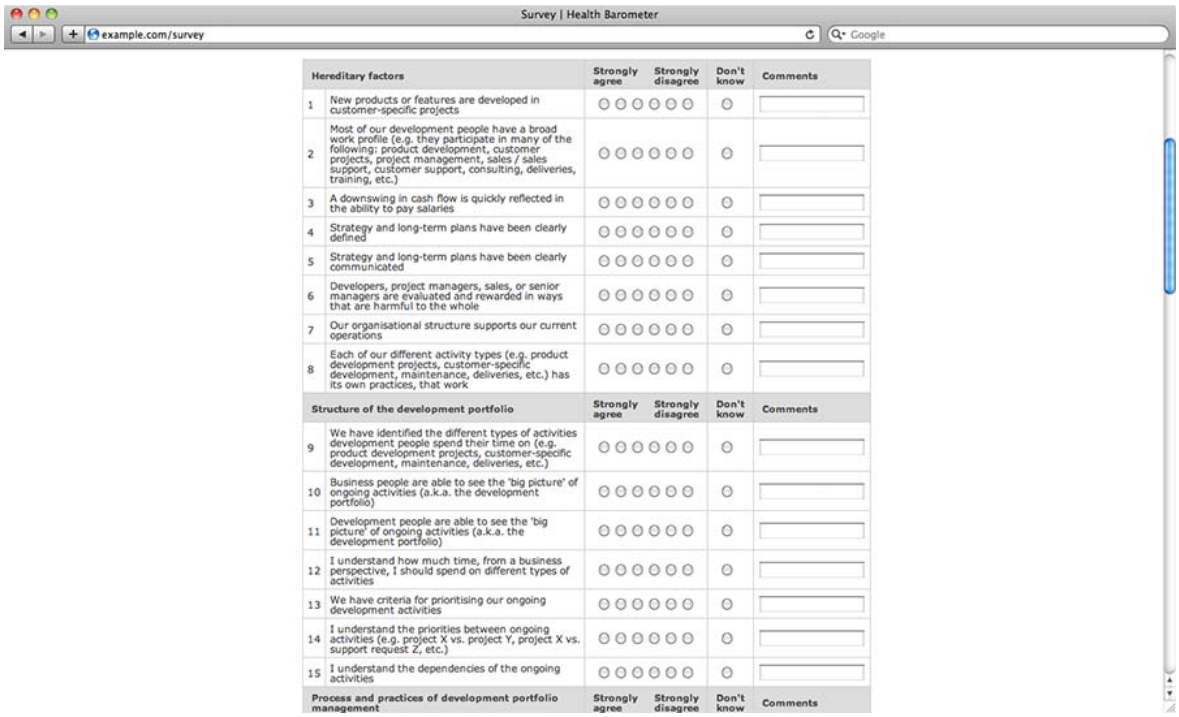
The usage of Scrum process model between the health barometer rounds was planned to be four sprints in length. The time period for this ranged from the beginning of December 2011 to the end of March 2012.

Once the fourth sprint was completed the second health barometer round was opened. The survey answers would again be collected from all employees and the same seven subjects would be interviewed. The survey was opened between 8th and 19th March 2012 and the interviews were conducted during the same time period.

In addition to the data gathered from the health barometer surveys and interviews also the data from Agilefant would be available in the result analysis phase.

The survey was conducted with the online survey tool provided by Aalto University Software Process Research Group. An example view from the survey tool is presented in

figure 4. The instructions and login credentials for the survey were given to the participants by email. The instruction email is presented in appendix A (in Finnish).



Hereditary factors		Strongly agree	Strongly disagree	Don't know	Comments
1	New products or features are developed in customer-specific projects				
2	Most of our development people have a broad work profile (e.g. they participate in many of the following: product development, customer projects, project management, sales / sales support, customer support, consulting, deliveries, training, etc.)				
3	A downswing in cash flow is quickly reflected in the ability to pay salaries				
4	Strategy and long-term plans have been clearly defined				
5	Strategy and long-term plans have been clearly communicated				
6	Developers, project managers, sales, or senior managers are evaluated and rewarded in ways that are harmful to the whole				
7	Our organisational structure supports our current operations				
8	Each of our different activity types (e.g. product development projects, customer-specific development, maintenance, deliveries, etc.) has its own practices, that work				
Structure of the development portfolio		**Strongly agree**	**Strongly disagree**	**Don't know**	**Comments**
9	We have identified the different types of activities development people spend their time on (e.g. product development projects, customer-specific development, maintenance, deliveries, etc.)				
10	Business people are able to see the 'big picture' of ongoing activities (a.k.a. the development portfolio)				
11	Development people are able to see the 'big picture' of ongoing activities (a.k.a. the development portfolio)				
12	I understand how much time, from a business perspective, I should spend on different types of activities				
13	We have criteria for prioritising our ongoing development activities				
14	I understand the priorities between ongoing activities (e.g. project X vs. project Y, project X vs. support request Z, etc.)				
15	I understand the dependencies of the ongoing activities				
Process and practices of development portfolio management		**Strongly agree**	**Strongly disagree**	**Don't know**	**Comments**

Figure 4: Example view from the online survey tool

4 CURRENT STATE ANALYSIS IN CASE COMPANY

The current situation of the software process and development portfolio management in the case company was analyzed by conducting a development portfolio health barometer round. Eight participants answered the survey and all developers and managers were interviewed. However one of the developers' interviews suffered from technical problems and could not be transcribed. This adds up to six interviews in total, consisting of three managers and three developers.

The case company had no formal process definition and were not collecting any velocity metrics or other numerical data, so the current situation analysis is based only on the results of the health barometer survey and interview. The results are divided into three parts similarly as in the health barometer: hereditary factors, lifestyle issues and symptoms.

All of the survey answers are presented as graphs. Based on the health of the individual statements the most critical ones are examined in detail by accompanying them with transcriptions of the interview answers.

A full listing of the answers to the survey is given in appendix B. The interview quotes provided in the following subsections are author's translations since the interview was conducted in Finnish.

4.1 Hereditary factors

The questionnaire results concerning hereditary factors are presented in figures 5 and 6. Figure 5 presents the questionnaire answers of developers and managers as a bar chart and figure 6 presents the median, the 1st quartile and the 3rd quartile of the answers as a radar chart. The numerical survey answers are converted to their textual representations as suggested by Heikkilä and Rautiainen (2010). The conversion is presented in table 5.

Table 5: Textual representation of hereditary factor median values

Number	Hereditary issue
1	Perfect
2	Good
3	Moderate
4	So-and-so
5	Predisposition for problems
6	The only option is to lead a strictly healthy life

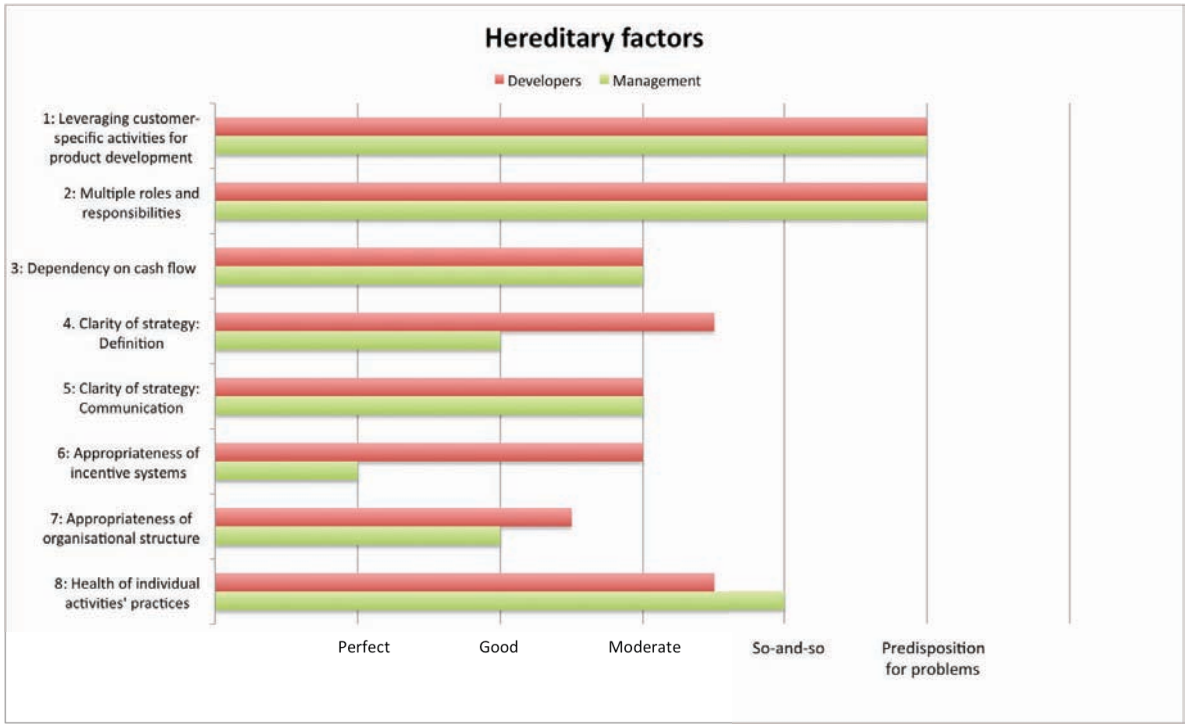


Figure 5: Hereditary factors in health barometer round 1, developers vs. management

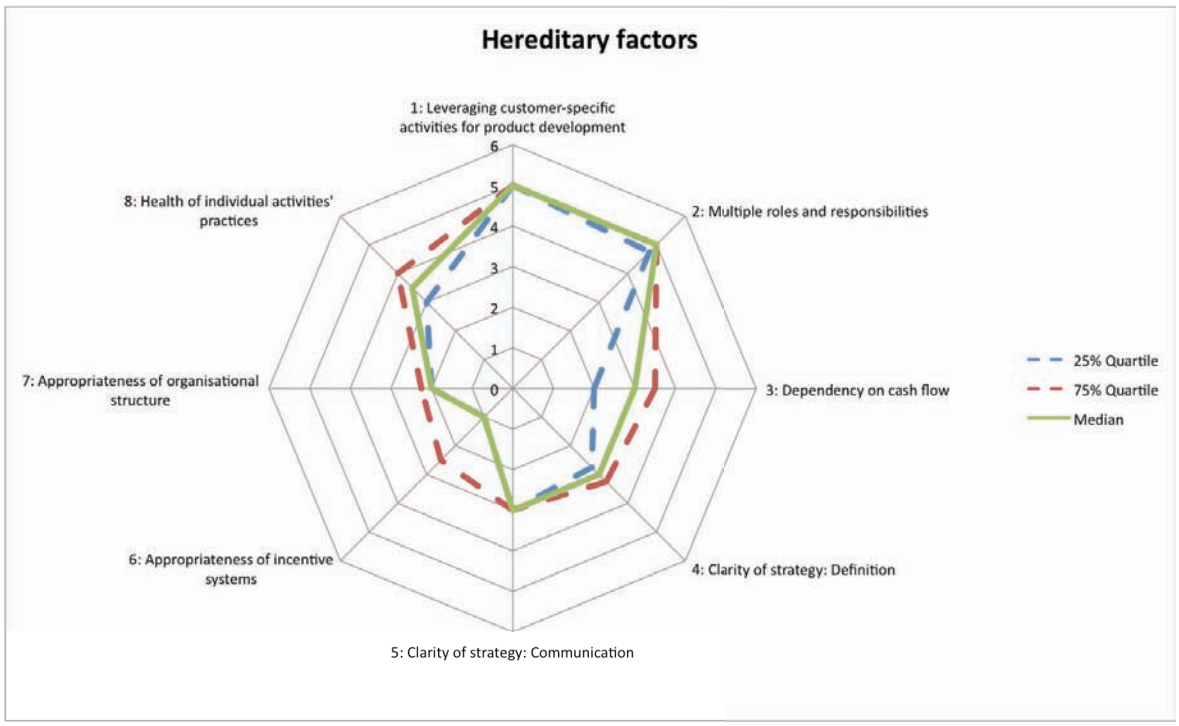


Figure 6: Hereditary factors in health barometer round 1, median and quartiles

As figure 5 shows both the managers and the developers agree strongly on statement 1 which is stated: *"New products and features are developed in customer-specific projects"*. Figure 6 shows that all in all there was a strong consensus about this statement as the interquartile range was 0.

Some quotes from the interviews concerning this statement are presented in the following:

> *"The changes we make for the software are customer-specific for business reasons and they are made upon customer request. Of course our aim is to implement the changes in such a way that they can be generalized and used by other customers as well. In the future our goal is to determine the direction of development more by our self. This would be much needed because the customers often lack the technical knowledge to make good requests."*

> *"We often start by implementing a quick version of a feature for a single client and then test it together with the client. If it proves to be useful we make a new version, which is part of a bigger set. Then the feature becomes available for other customers as well."*

> *"A lot of things start with a customer asking about a feature. Then we make a pilot project where the customer is able to test it. In the end it often ends up being a new feature or a product that is also suitable for other customers. The goal is a generic solution but usually in the beginning it's customer specific. There are other examples as well but this is the common case."*

Figure 5 also shows that both managers and developers agree on statement 2: *"Most of our development people have a broad work profile (e.g. they participate in many of the following: product development, customer projects, product management, sales/sales support, customer support, consulting, deliveries, training etc.)"*. As figure 6 shows, the interquartile range is again very small, only 0,25.

The following quotes are again from the interviews:

> *"Lately we have luckily been able to add the amount of workforce a bit which has helped to reduce the amount of "hats" a single employee has. You don't have to master all possible development skills any more but a lot of different responsibilities still remain. It often causes a bottleneck when a lot of work piles up for a single person. On the other hand it's also a good thing that the work is versatile, it keeps things interesting."*

> *"This is the way it is in a small organization"*

> *"Everyone has to participate widely. But recent development has enabled us to differentiate production from sales and marketing and also the customer support manages to handle things more and more without the help of developers."*

> *"At the moment the size of the development team is so small that everybody has multiple "hats". The goal is to enable more specialization in the future but also to maintain rotation in the work responsibilities."*

Especially management seems also to be concerned about statement number 8: "*Each of our different activity types (e.g. product development projects, customer-specific development, maintenance, deliveries, etc.) have their own practices, that work*". The median answer for this statement was 3,5 and interquartile range 1, but management median 4.

4.2 Lifestyle issues

The results of lifestyle issues are presented similarly as the ones in the previous subchapter. Figures 7 and 9 present the median answers of management and developers compared with each other as bar charts and figures 8 and 10 the total median answers together with the 1st and 3rd quartiles as radar charts. The numerical answers are converted to their textual representations as presented in table 6.

Figure 7 shows a clear trend of developers being more pessimistic than the managers. In the developers' opinion the most critical issue here is statement 13: *"We have criteria for prioritizing our ongoing development activities"*. The median value for this statement is 4 and developer median 4,5 but manager median only 2. As figure 5 shows the interquartile range is 2, which further indicates that there is disagreement on this topic. Here are some quotes from the interviews with managers:

> *"No written criteria exist. A spoken one does and it is often reviewed in product development meetings: have the priorities changed or not. This has not been communicated clearly enough, but it is pretty clear anyway. We do certain things for the large customers and the product development depends on this. There are also some strategic product development efforts that aim to strengthen our position on the market. These are the important ones. Lets understand my answer so, that we have the criteria but it is not written down or communicated well enough."*

Table 6: Textual representation of lifestyle issue median values

Number	Lifestyle issue
1	Exemplary
2	Got it covered
3	Reasonable
4	So-and-so
5	Clearly room for improvement
6	High risk and ready to crash

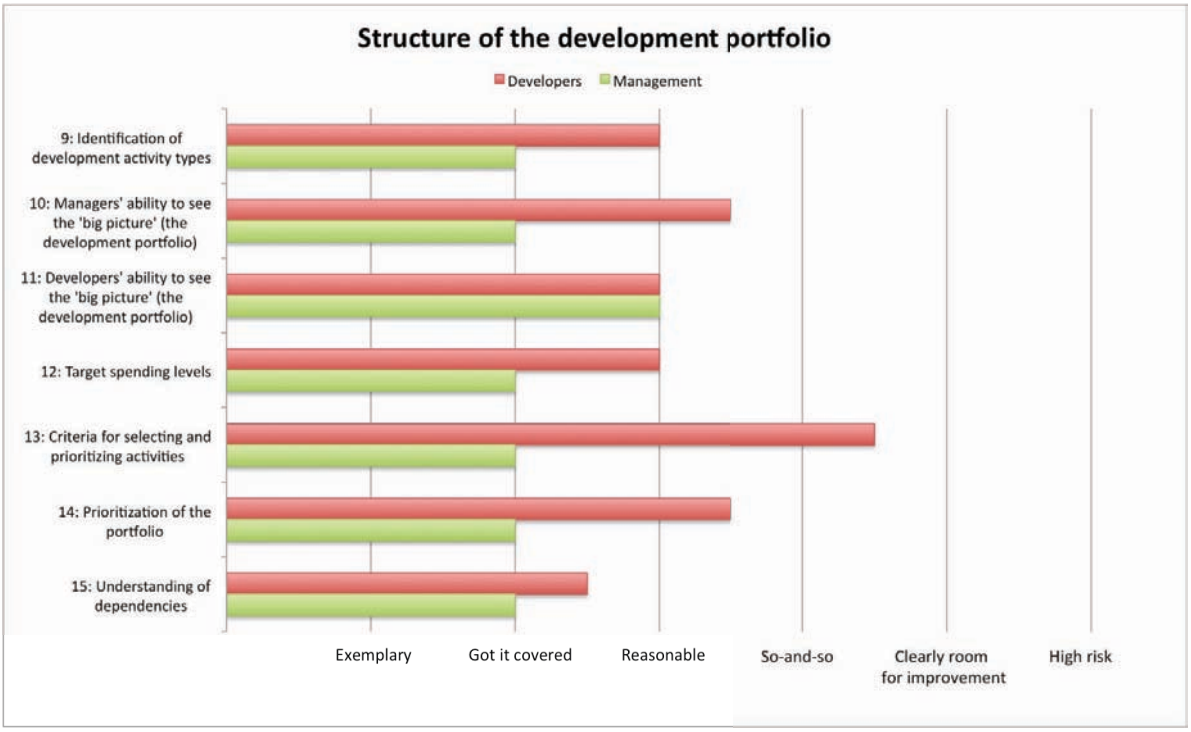


Figure 7: Lifestyle issues part 1 in health barometer round 1, developers vs. management

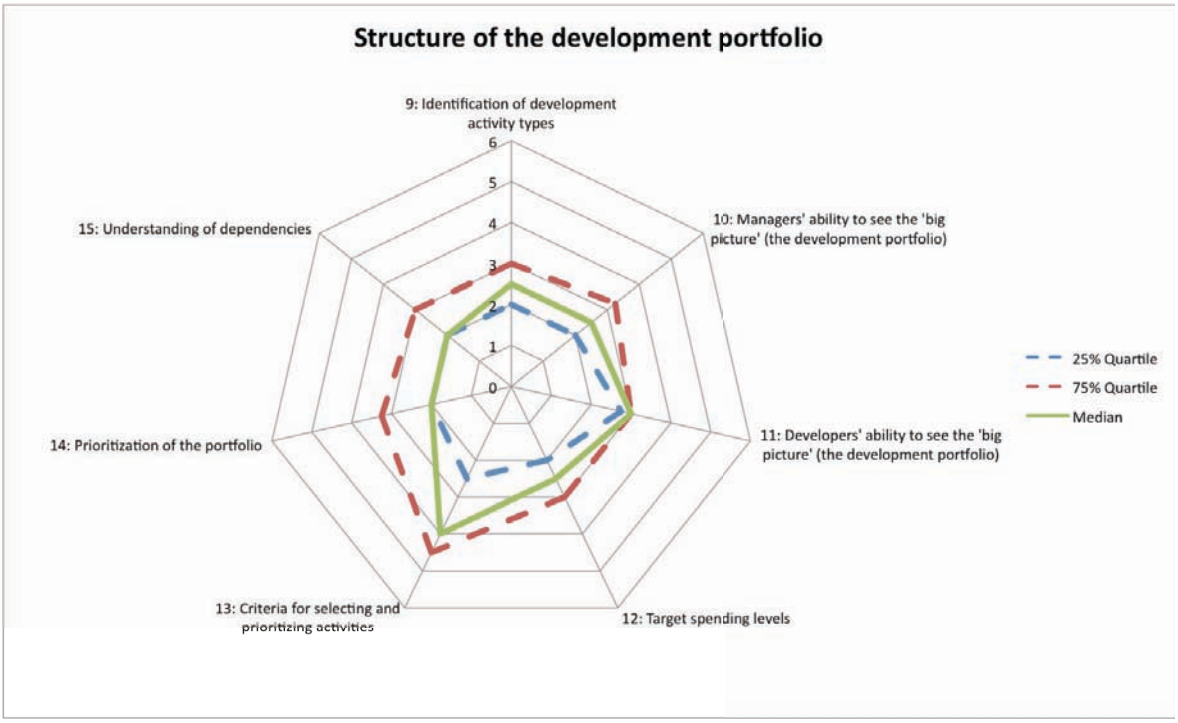


Figure 8: Lifestyle issues part 1 in health barometer round 1, median and quartiles

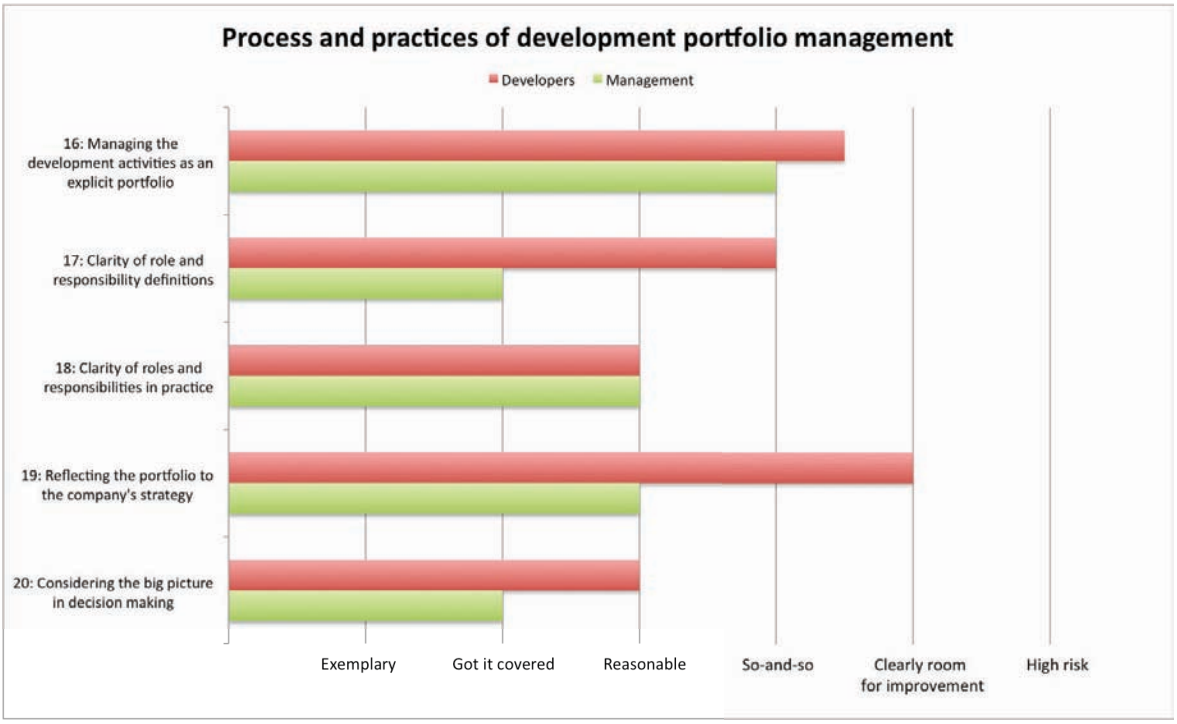


Figure 9: Lifestyle issues part 2 in health barometer round 1, developers vs. managers

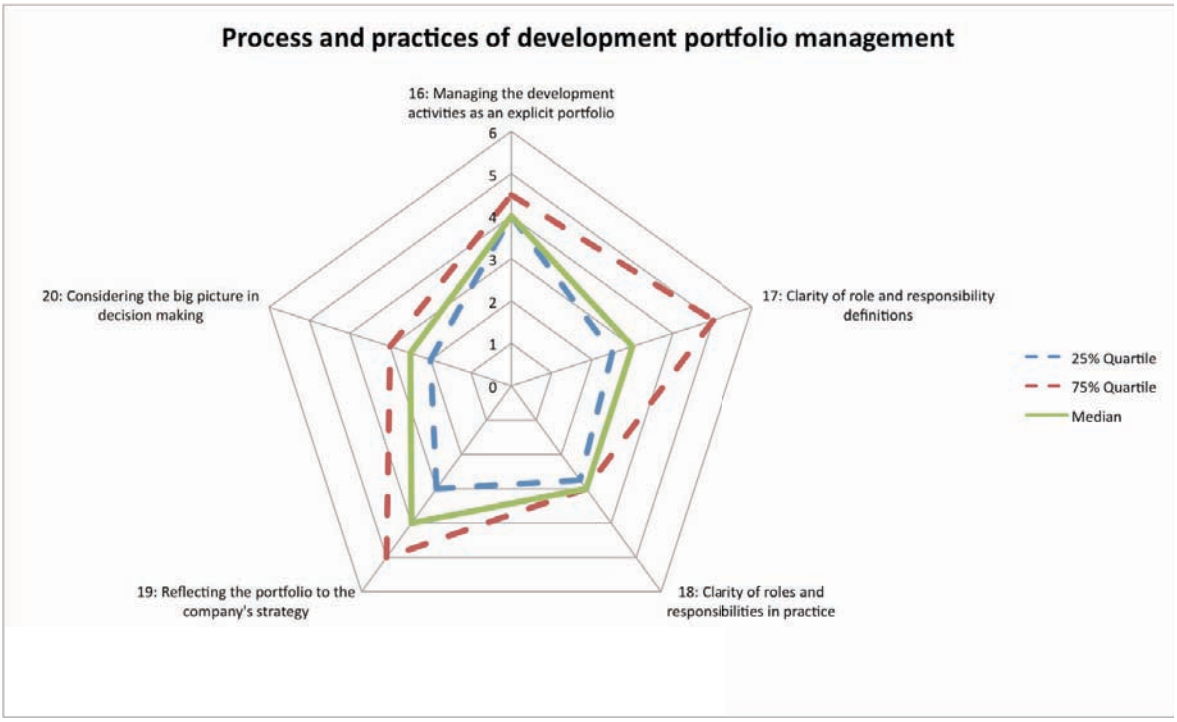


Figure 10: Lifestyle issues part 2 in health barometer round 1, median and quartiles

> *"Some kind of implicit criteria exists and it originates from customer needs which are the things that bring money to our bank account. The criteria have not been formalized but strategically important things have been recognized. When something new comes on the table it is evaluated based on these."*

The following quotes are from the developers' interview answers:

> *"This is connected to some of the previous questions. The management is responsible for the prioritization and if they don't understand the big picture totally then it is difficult for them to prioritize or at least to communicate it. And because of this I don't really know how they like my time management. If everybody is a bit unsure about the topic it is difficult to ask for help or clarification. And it is not useful to have lengthy discussions about the same topic every week, we should have a more formalized method for prioritizing. Someone makes a decision which is than clearly visible for everybody."*

> *"There is nothing written. I guess an unspoken consensus exists. If I have to choose between two things I sometimes make the decision my self and sometimes ask from management. It is also important to be able to affect the order on the lower level by your self, otherwise the work becomes too micro-managed."*

> *"I consider the prioritization to be one of the Achilles heels. I'm sure others will also say that stuff often appears from nowhere, also urgent work. And this is often stuff that has been known before, but not communicated earlier."*

In figure 9 the same trend about developers being more pessimistic is again visible but less strikingly than in figure 7. Developers see statements 16, 17 and 19 as problematic and only with statement 16 the managers agree.

Statement 16 -*"All the ongoing and immediately upcoming development activities that require attention from the developers are managed as an explicit portfolio"* - got a median value of 4 while the interquartile range was 0,5. The following quotes from the interview were found interesting:

> *"The distribution of responsibilities between the managers is often quite confusing. It is difficult to know who to ask for clarification and when. It is possible that it is managed more explicitly as a portfolio that it seems from the developer's point of view but it just isn't communicated so well. The important thing would be to have a single place where all projects,*

deadlines and other work items would be clearly listed. It would help tremendously to understand the bigger picture."

"At the moment the management method is simple: meetings where all managers and developers are present. But there is a clear need for better methods and tools."

Statement 17, which states: *"We have defined who are responsible for development portfolio management"*, was the most disagreed statement in the questionnaire. Median value was 3 but the interquartile range 2,5. The management considered it to be well covered with a median answer of 2 but the developers gave a median answer of 4. Here are some interview quotes to explain the answers:

"We know who these persons are and I think we might have something written down in the wiki. This is absolutely changing now when the amount of employees is rising."

"We are not good at this. The definition of responsibilities is more or less still open. Normally we meet up with everybody who is available at the moment and try to find a consensus."

Also statement 19, *"We actively reflect the content of the development portfolio to the company's strategy"*, was disagreed. The median value for all was 4 and interquartile range 2 but the median for developers was 5 and for managers 3. The interview revealed the following:

"Sometimes we discuss strategy, but I don't remember that we'd clearly evaluate how the daily development work reflects to the long term plan."

"Some kind of general plan exists and it is updated while new information becomes available. It feels like the reflection only works in one direction: we only think about the next steps, what is needed next, and act accordingly. We don't pause afterwards to think if the decision was right or not. I'd hope that this kind of reflection was done all the time on the management side."

4.3 Symptoms

The results regarding symptoms are presented in three parts because of the large amount of statements in this category. Figures 11, 13 and 15 present the median answers of managers compared to developers and figures 12, 14 and 16 the median answers together with the 1st and 3rd quartiles.

Figures 11 and 12 show that both developers and management agree that the statements 23, 24, 25 and 35 are problematic. These statements all revolve around the same resource sufficiency theme.

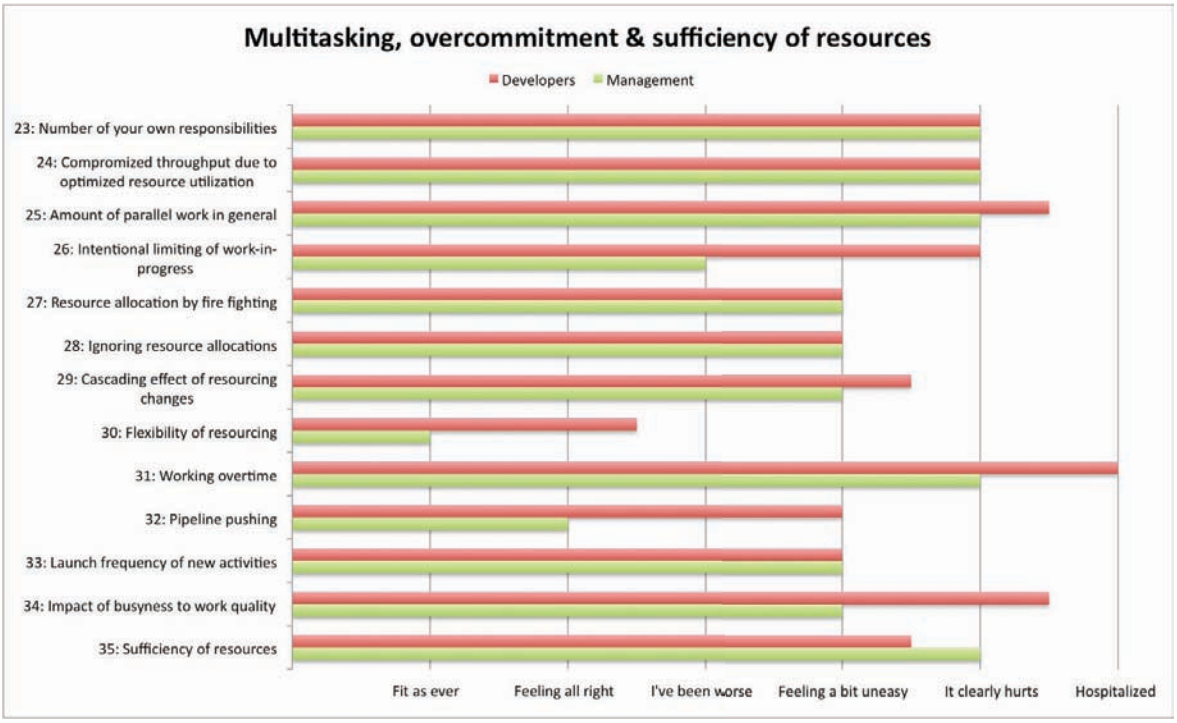


Figure 11: Symptoms part 1 in health barometer round 1, developers vs. management

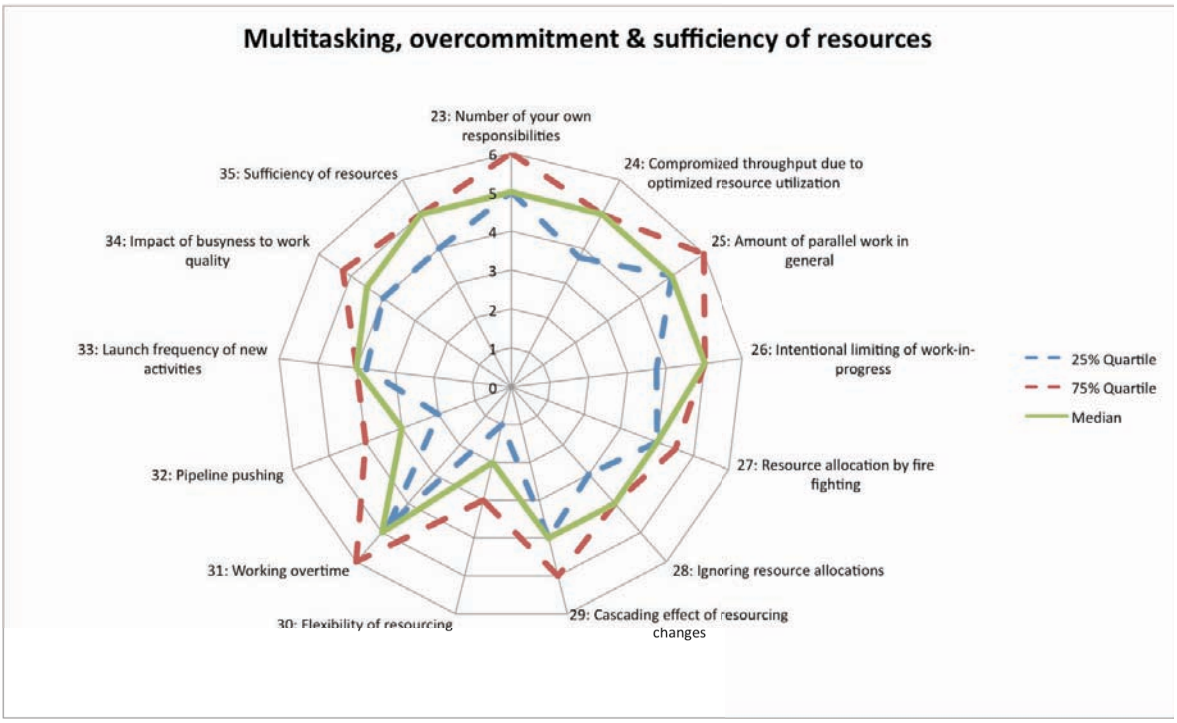


Figure 12: Symptoms part 1 in health barometer round 1, median and quartiles

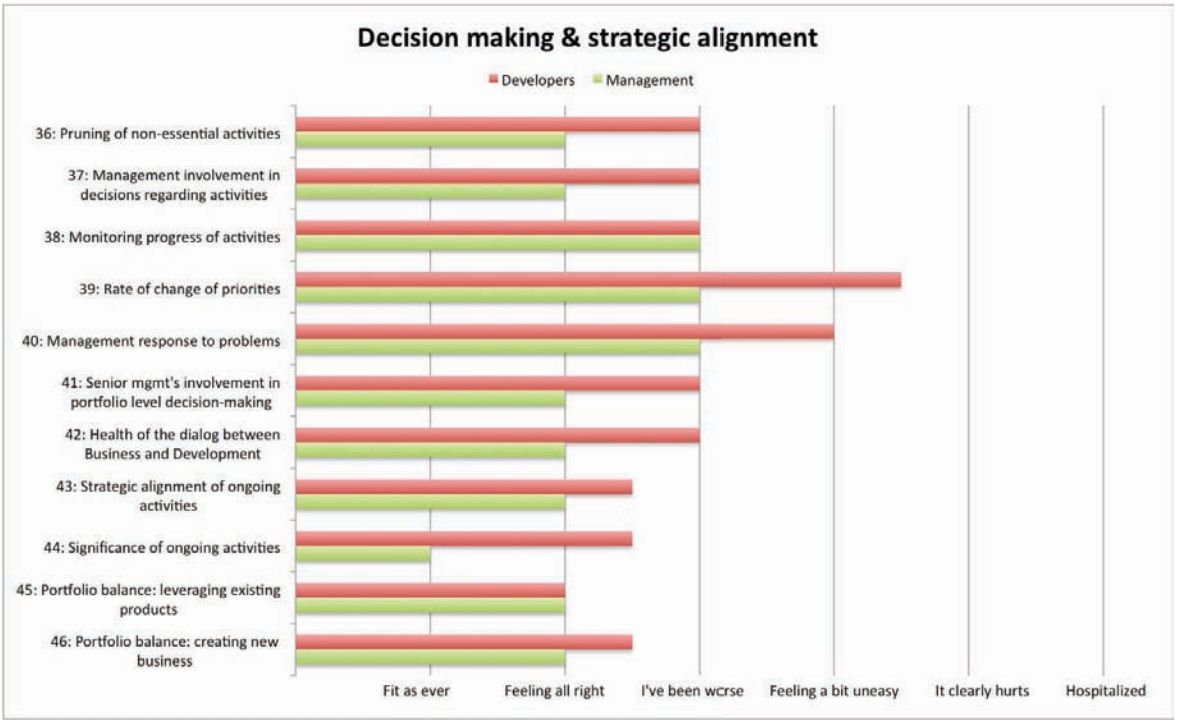


Figure 13: Symptoms part 2 in health barometer round 1, developers vs. management

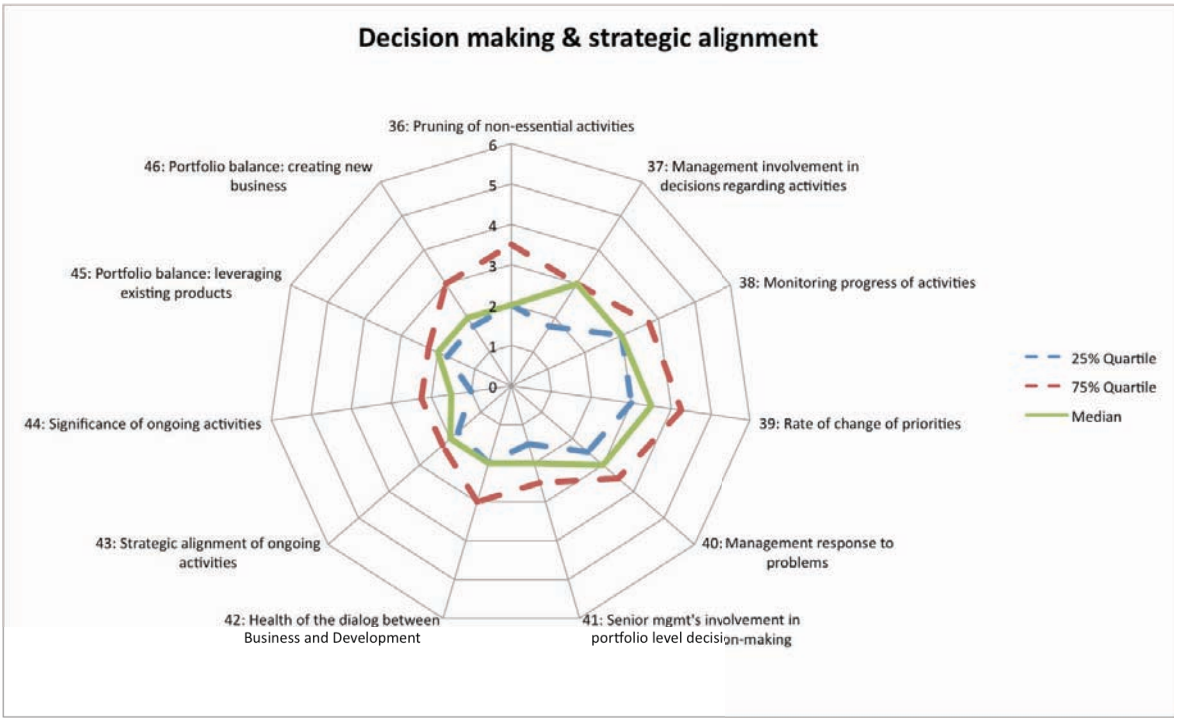


Figure 14: Symptoms part 2 in health barometer round 1, median and quartiles

Statement 26, which states: *"We complete one thing at a time and don't shift our attention from one incomplete task to another"*, had again more disagreement. The median value for developers was 5 but for management only 3. Here are some interview quotes:

> *"We've had improvement with this lately. We don't shift our attention away from the planned activities because of some minor issues any more. But it still happens that some bomb explodes and all work planned for the next week is moved into future and we start doing something else"*

> *"We just spent some time listing not started and incomplete things that needed to be finished and found out that the list of 'almost complete' stuff was rather long. The only things missing were often just phone calls, emails or some really small fixing. This suggests that we would benefit from a more rigorous process where things would be done one by one until complete. This is something we need to focus on."*

Both developers and managers agreed that they are working overtime, although it was not considered a big problem. In statement 34, *"Our employees have too much work to do and quality of work suffers from it"*, the median answer of managers was 4 but the median answer of developers was 5.5. The interview answers contained the following statements:

> *"The problem is clearly visible. The situation is not really bad, but problematic anyway. On the other hand, if we consider the fact that our current products lack sufficient unit tests, I could have given the worst possible grade. So yes, the quality has suffered by not having enough tests."*

> *"We work on a feature until it functions properly and stop there. It depends on the developer doing the work how well the end result functions and how controlled the process to get there is. But after that the tests, code reviews and documentations are usually nonexistent."*

The statement 39, *"The priority ranking of activities changes constantly"*, caused some disagreement. The median value was 3.5 but the interquartile range was 1.25. Most employees didn't perceive this as a big problem but some of the developers did. Here are some interview quotes concerning this statement:

> *"There is always some surprising work that comes up."*

> *"The word 'continuous' in the statement is tricky – what does it exactly mean? The issue most affected by priority changes is the development of our next generation product. It seems to often be the pushover although it is extremely important. In practice it is the first activity to suffer when the resources run scarce. This issue crystallizes the tradeoff between future product development and currently accute customer projects."*

From figure 15 it can be seen that developers find statement 47 problematic and both developers and management agree with statements 51 and 52. Statement 47, which says that "*Ongoing activities are behind schedule*", has a developer median of 6 and a management median of 4. The high median answer for developers is partly explained with the fact that one of the four developers used a "don't know" answer in this statement, which translates into a blank answer in the analysis. Anyhow it seems that developers are much more worried about the progress of ongoing activities. Here are some interview quotes from the developers:

> *"Something always gets done in time, so the promised deployment deadline is often met. But if we wanted to do it well - with tests and everything – then we are not up with the schedule."*

> *"We often have to bargain with the tests. Regarding the new product I haven't had time to do much. This is also because the work piles up to a single person, for example with the new product I have been waiting for another developer to complete some of the requirements specification but he is so busy that the work is not proggressing. This can be explained with all the surprising urgent work that comes up from nowhere."*

The management answers included statements like:

> *"The projects that we are currently working on are mostly on schedule. In addition to these we have some stuff that do not have a fixed timetable and they are somewhere on the background. There is maybe too much of these things. But the projects we have decided to do are quite well on time."*

> *"Especially considering the next generation product I could have agreed even more than in the answer I gave. The urgent daily things and the needs of the paying customers have often pushed it aside and I have to admit the situation has given me some headache."*

The statements 51, "*We should invest more in improving the practices of individual activities (e.g. project mgmt., team practices, deployment processes, sales processes, customer support, etc.)*", and 52, "*We should invest more in improving development portfolio management (e.g. prioritizing activities, linking strategy with daily work, structuring the development portfolio, etc.)"*, both had a median answer of 5 and the interquartile ranges were 0,25 and 0,5. No large difference between developers and management was found. The interview answers included the following:

> *"We need to improve both [individual activities and portfolio management], I don't think improving just one is enough."*

"We need to improve by making the management work and use it to fix the individual activities. Big picture first so that we're able to focus on the smaller things."

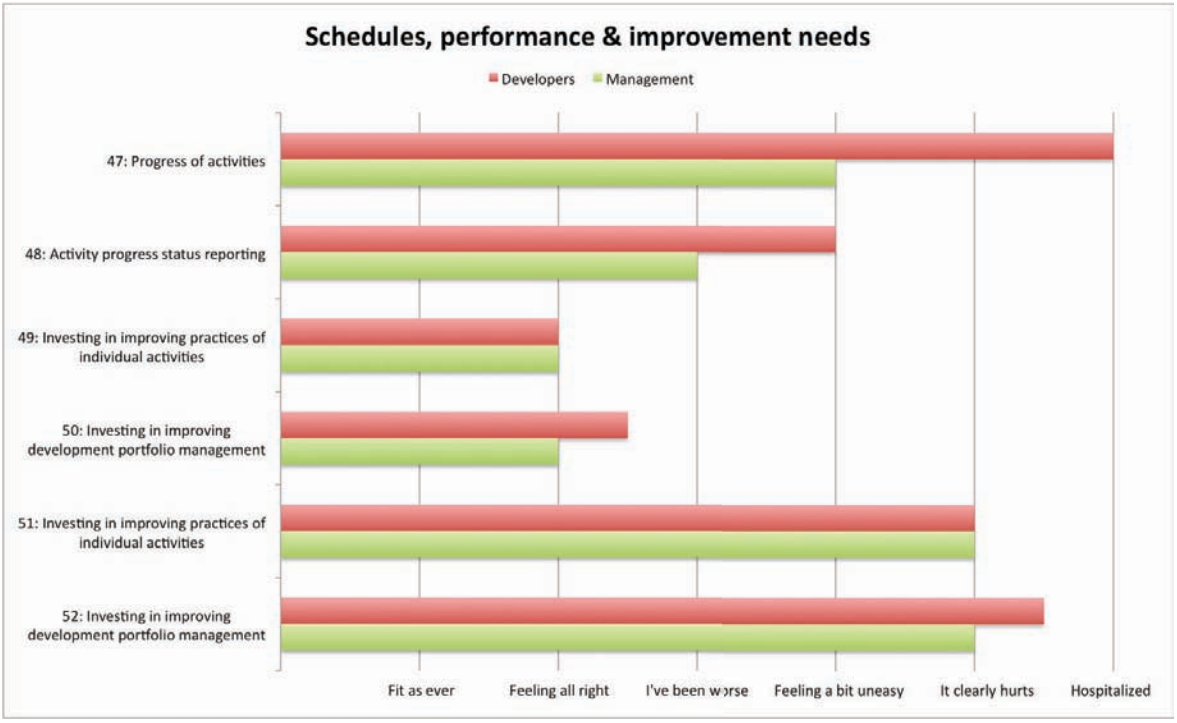


Figure 15: Symptoms part 3 in health barometer round 1, developers vs. management

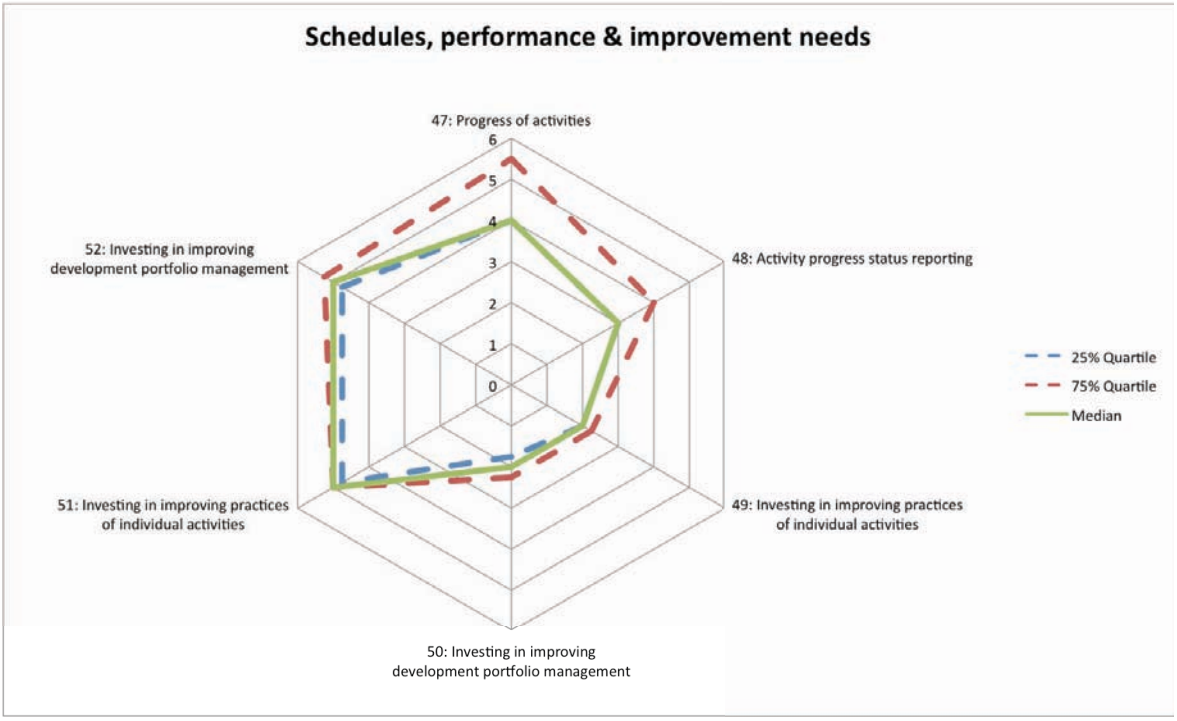


Figure 16: Symptoms part 3 in health barometer round 1, median and quartiles

4.4 Overview and analysis

From the statements concerning the hereditary factors three issues were found more noteworthy than the rest:

1. Product development can be described as customer specific
2. The employees have multiple roles and responsibilities
3. Especially developers are unhappy with the health of some of the individual activities' practices

By analyzing the interview answers it can be argued that the customer specific manner of product development is a conscious decision, possibly even a stategy. Since the company is self-financed without external funding it is natural to focus on projects and software features that have a confirmed buyer already before development is started. However, this makes it important to focus on prioritization in order to be able to make informed decisions about when to produce work only for customer projects and when to focus on self-specified product development.

The organization is small with only eight employees, which makes the second point seem natural. With only four developers it is necessary for all to have multiple roles and responsibilities. But with this situation it is important to have well defined and efficient practices for different development and management practices. It can be assumed that the company would benefit greatly from improving on the third point.

The challenges found in the lifestyle issues include the following:

1. The criteria and methods for selecting and prioritizing activities are somewhat unclear
2. The development portfolio management lacks in tools and definitions
3. Roles and responsibilities are not clear at least in the developers' opinion
4. The amount of reflecting the development portfolio to company's strategy could be increased
5. In general the developers found the lifestyle situation more problematic than the management

The first finding resonates with the hereditary issue of having customer specific product development. If the customers' needs conflict with the companys own product development plan the prioritization can be complicated. It can lead to frequent changes in plans and priorities, which becomes confusing especially to developers not having a good understanding of the business situation.

The third point can be seen as a consequence of the fact that the organization is small and the employees have multiple responsibilities especially when, as the second point suggests, the tools and definitions of development portfolio management are in need of

improvement. Also the fact that the reflection between development portfolio contents and company strategy is found insufficient could suggest that the portfolio could be better defined. The fact that developers experience the lifestyle situation more problematic than the managers could result from the managers not being fully aware of the proggress of development activities.

The symptoms found in the health barometer can be grouped under the following themes:

1. Resources are limited and it is displayed at general busyness at work
2. A single person has multiple roles and responsibilities, which makes it difficult to consentrate on a single work task at a time without interruptions
3. The feeling of busyness is further increased by frequent changes of plans and priorities
4. The quality of work suffers as the result of the previous issues
5. The activies appear to be behind schedule
6. There is a perceived need to improve both development portfolio management in general and individual activities' practices

All of the experienced symptoms seem to connect well with the findings from hereditary and lifestyle issues. Since the employees have multiple roles and responsibilities and there are challenges with prioritization, development portfolio management and definitions of responsibilities it seems only natural that especially the developers feel that resources are limited and work is always busy. The important question here is could the same amount of employees produce the same work output with better quality and less stress if the development portfolio management and individual activities' practices were improved?

The last point shows that both the managers and developers have observed the need to improve the way the company is working. This can be seen as a good starting point for process improvement activities because it suggests that motivation exists.

5 IMPLEMENTING AGILE PROCESS IN THE CASE COMPANY

This section presents the adoption of the new process in the case company, which was done between the two health barometer rounds. The first subsection analyzes the feasibility of Scrum process model for the case company, the second subsection presents the details of Scrum implementation and the third subsection explains how the case company kicked off the new process.

5.1 Is Scrum the right answer?

Based on the findings presented in the previous chapter it is possible to evaluate the usefullness of Scrum for the case company. If we first investigate the operating environment and the hereditary factors of the company it can be stated that Scrum fits the size of the company and its development team.

The issues the employees experience with prioritization, busy schedules, role definitions and ability to consentrate are all possibly relieved by efficient usage of Scrum. Scrum helps to define roles clearly and makes it easier to keep everyone up-to-date about the current priority order of things by using a backlog. The time pacing trough sprints and daily Scrum meetings could help the developers in completing a single task before switching to the next. Scrum also provides mechanisms for reflecting both the work done and the process, thus enabling continuous improvement also in the future.

The main challenges in deploying Scrum with the case company included the following:

- The four developers had somewhat specialized skill sets, which meant that not everyone could do all types of tasks
- The maintenance and technical support work of the two production systems was conducted by the same developers as the actual development work, which meant that the unplanned interruptions caused by urgent maintenance and technical support would interfere with the learning process
- The product owner role was previously distributed for multiple managers
- The CEO of the company, who had previously also acted as the principal product development manager left the company during the implementation period, which affected especially the product owners

5.2 Details of Scrum implementation in the case company

The selected method for communicating requirements was the form of user stories. This is beneficial for the case company, because the user stories have a non-technical approach for

requirements management and at the case company the employees that spend the most time with clients are responsible for non-technical work such as sales, marketing or customer service. By using user stories it is easier for these employees to communicate the observations they make while interacting with actual users of the developed software.

The products and projects in the case company were modelled into the Agilefant software so, that the two products currently in production state were both given their own product backlogs. Also the future version of the other product was modeled as and independent product backlog. Additionally, an outsourcing project conserning IT-infrastucture was separated into its own product backlog. Finally a separate product backlog was created for internal work, such as meetings, holidays, daily repetitive administration tasks et cetera. In total this adds up to five product backlogs.

The Scrum master role was given to the author of this study and the product owner role was distributed among three persons. They were given one or two products out of the five to manage.

It was decided to conduct the daily Scrum meetings via chat software the company uses heavily during home office days. During office days the meetings would be held normally with everyone in the same room.

The goal in the process definition phase was to start using Scrum without unnecessary delay by selecting widely adopted and industry-proven methods and not over-analyzing the situation beforehand. Once the team would be using Scrum, they could use its built-in iterative nature and retrospective sessions to make corrections and improvements that are customized for the case company and its working environment.

5.3 Deployment of Scrum process

The implementation phase of the process improvement started with a product backlog preparation meeting. The product owners of the different products were present together with other key stakeholders.

During the meeting the work-items from the current process were transformed into user stories and entered into the Agilefant software. The stories were refined and split if necessary and an initial prioritization was conducted.

Before the actual kick-off a second meeting was held where all developers and other stakeholders were present. The basic consept of scrum was familiar to the employees since the case company had previously experimented with Scrum. However, a short introduction to the basic principles, roles, events and definitions of Scrum was given which turned out to be useful.

The schedule for the first four sprints is presented in table 7. The gap between the first and second sprints was caused by the Christmas holiday season. The day between iterations is used for the demonstration, reflection and planning of the next iteration.

The actual usage of Scrum in the case company begun with a sprint planning meeting on December 4th. The first sprint planning started by using the morning to make story point estimations and continued in the afternoon with the detailed effort estimations. The following sprint change days were started with the demonstration of the previous iteration and the reflection session followed by the planning.

Table 7: Schedule of the first sprints

Sprint no.	Start date	End date	Length (workdays)
1	December 5th 2011	December 21st 2011	13
2	January 5th 2012	January 25th 2012	15
3	January 27th 2012	February 15th 2012	14
4	February 17th 2012	March 9th 2012	16

6 RESULTS

This chapter presents the main results of this study. The results are reported in two subsections: first the data built up by using the Agilefant software to manage Scrum process during four sprints, and secondly the analysis of the second development portfolio health barometer round.

6.1 Sprint metrics

The main metrics from the sprints completed within the observation period of this study are presented in table 8. The table shows the length of the sprints measured in full workdays, the amount of story points completed in the sprints, the amount of estimated effort completed in the sprint and the amount of points and effort completed on average per day. The metrics show that the amount of points and effort completed per day was increasing steadily throughout the whole period. This can be caused by actual performance increase or by increase in the accuracy of effort estimation. In both of the cases the effect is positive for the company.

Figure 17 presents example burndowns of all four sprints. The product in question is one of the two products currently in production state. The first burndown (top left) shows that originally the sprint was planned to include far less work than it ended up containing. Three scope changes, which are visible in the graph as upward vertical dotted lines, were committed and the nature of the sprint changed a lot. The first scope change was done only three days after the sprint started, which suggests that either something surprising came up or that something was not taken into account during the sprint planning.

Table 8: Main metrics from completed sprints

Sprint	Length (workdays)	Points completed	Points per day	Effort	Effort per point	Work per day
1.	13	54	4.2	236h	4.4h	18.2h
2.	15	93	6.2	319h	3.4h	21.3h
3.	14	106	7.6	325h	3.1h	23.2h
4.	16	117	7.3	380h	3.2h	23.8h

The following three sprint burndowns look much better as they do not include multiple scope changes. Only sprint number 3 includes a single addition to the plan created in sprint planning. This means that the team has had a better chance of consentrating in getting the work at hand done before switching to the next task. The burndown of sprint 3 also includes an outward scope change. This is a good sign since it suggests that the product owner has understood the basic principle of commitment to the contents of a Scrum sprint: once the contents are fixed during the sprint planning, all work added to the plan demand removing something else so that the team workload stays the same. This also forces the product owner to think carefully about the priorities before committing changes to the plan.

Because the complete workload distributes between the five products unevenly the example burndowns do not tell the complete truth about the success of the sprints in the other products. This has to be kept in mind when making conclusions about the success of Scrum implementation in the case company. Some scope changes were made in the other iterations even in the fourth sprint and discussions during the demonstration and reflection sessions showed that the transition from an ad-hoc process into Scrum was not painless. However, the numerical results presented in table 8 give evidence about the benefits of the process improvement actions.

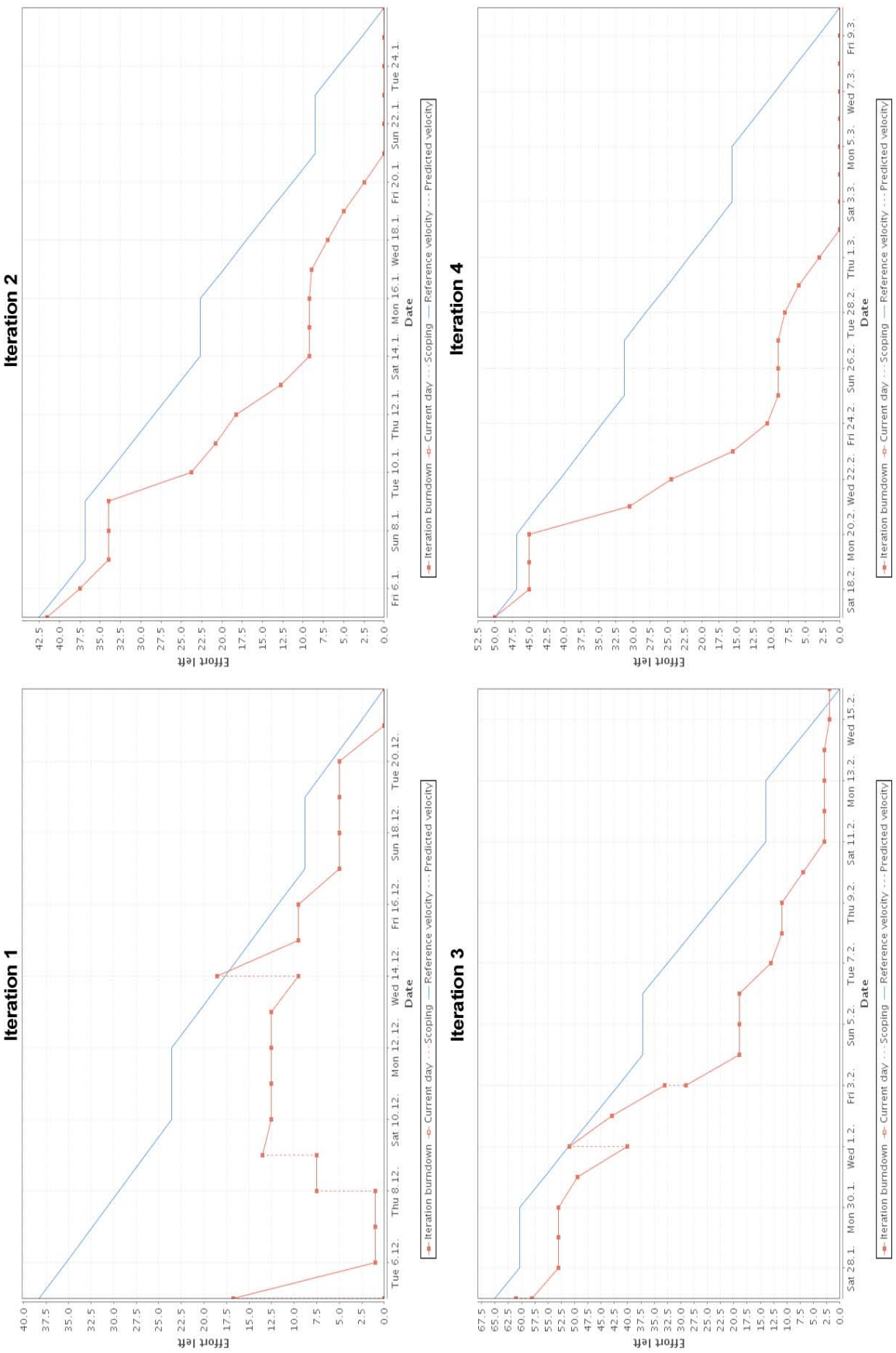


Figure 17: Example sprint burndowns

6.2 Second health barometer round

The second health barometer round including the survey and the interviews was conducted between March 8th and 19th 2012. All eight employees answered the survey and the same seven as on the first round were interviewed. The only change was that the CEO of the company changed in the beginning of year 2012. This means that one of the three managers answering the survey changed between the two rounds.

The full listing of numerical answers for the survey are presented in appendix B. As explained in section 5, the interviews were conducted in Finnish so also the quotes presented in this section are author's translations.

6.2.1 Hereditary factors

Figure 18 shows the answers of developers and managers to the hereditary factors section of health barometer round 2. Figure 16 presents the median answers of the two rounds compared to each other.

Figure 19 shows that the three hereditary factors considered most challenging on the first round have all experienced a slight change in the positive direction. Especially the statement number 2, "Most of our development people have a broad work profile", is now perceived as a less critical issue. The interview answers concerning this statement included the following:

> *"Now that we have been doing stories I have a feeling that work has centralized a bit. It might be that it has happened only to me."*
>
> *"I feel like it has narrowed down a bit, I mean in a positive way. This is however only my feeling, at least for me it seems less broad."*
>
> *"It has improved a little. We have been able to separate support issues from messing up with actual development work."*

It seems that the amount of roles and responsibilities for the developers have reduced or at least the amount is perceived to have reduced. This could be due to the fact that the different stories and tasks are assigned to different developers during the sprint planning. This way it is possible to control the individual workload of a single developer, not only the workload of the team as a whole.

Also statements 1 and 8 that were classified as challenges during the first round now have a smaller median answer. The differences are small but together with the fact that none of the eight statements in the hereditary factors category had a bigger median score during the second round they show a positive trend.

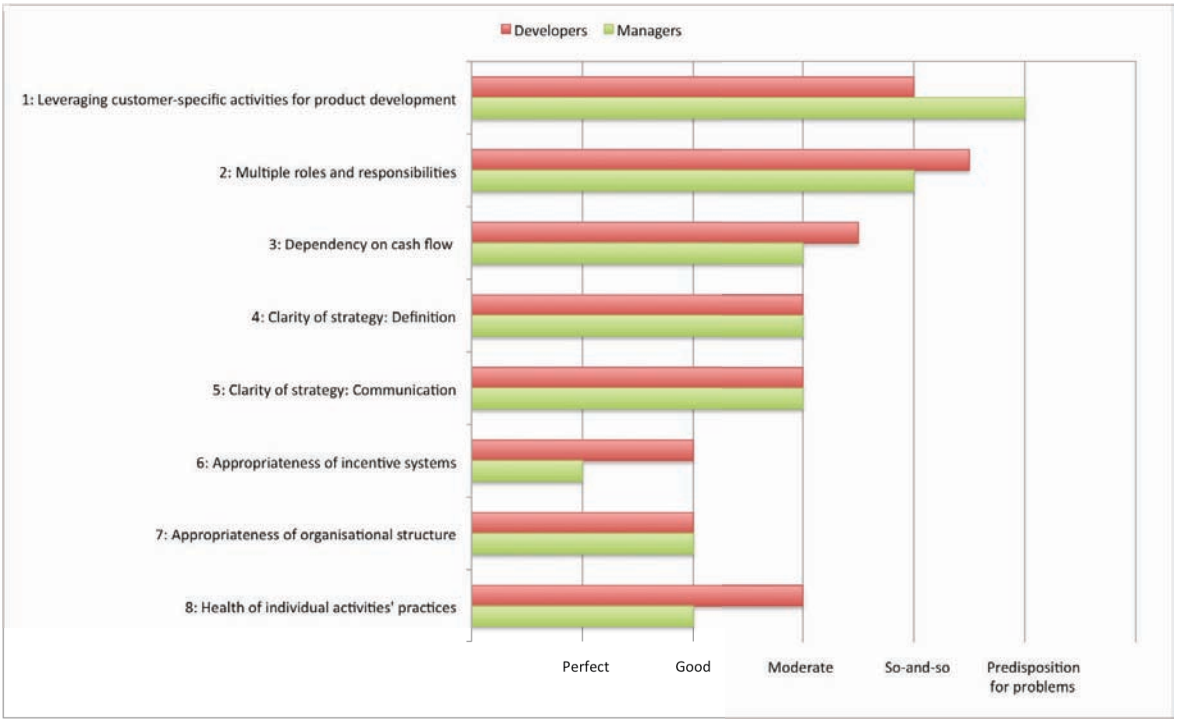


Figure 18: Hereditary factors in health barometer round 2, developers vs. managers

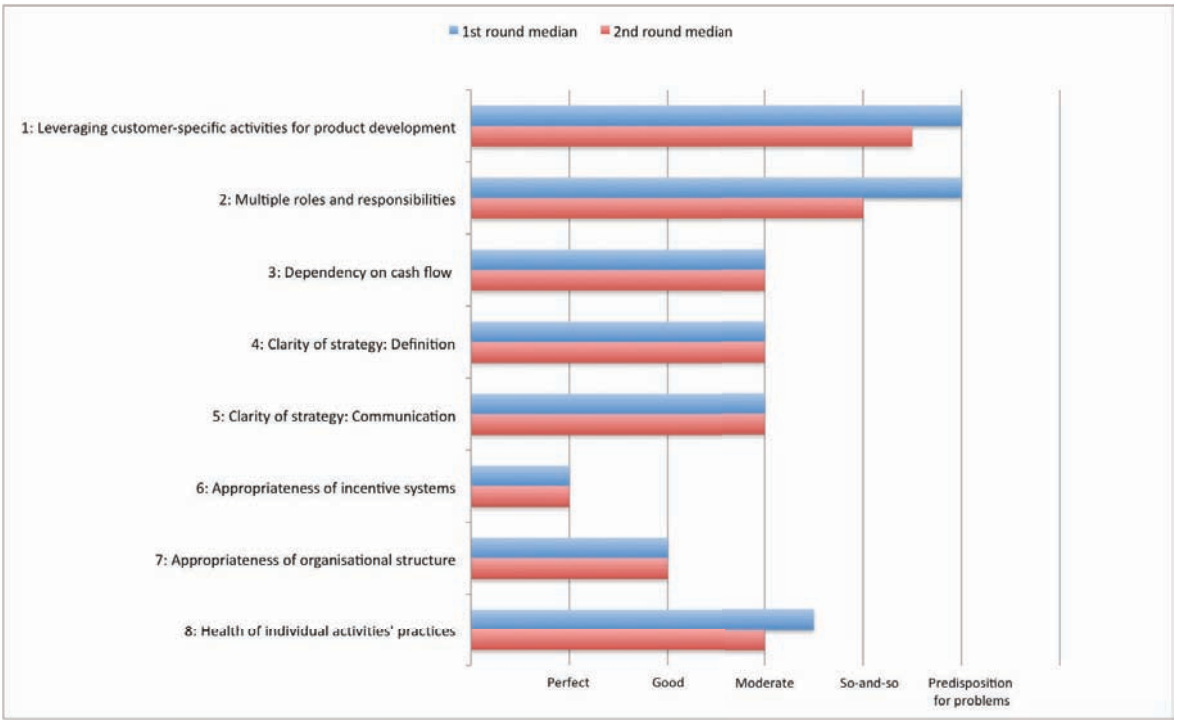


Figure 19: Comparison of hereditary factors between rounds 1 and 2

6.2.2 Lifestyle issues

Figure 20 displays the developers' and managers' median answers to lifestyle issues during the second health barometer round and figure 21 gives a comparison of the answers to lifestyle issues between the first and the second round.

The difference between the rounds is clearly visible in figure 21. Almost all of the median answers have decreased and especially the statements 13 and 16 have improved substantially.

The statement number 13, which reads: "*We have criteria for prioritising our ongoing development activities*", had a median of 4 during the first round but has now reduced to 3. Here are some interview answers to further explain the changes:

> *"The criteria itself has maybe not changed but now we are forced to decide beforehand which items to work on next and which to move forward."*
>
> *"Now that we are planning things ahead at least on a sprint-level things have improved. The middle- and long-term planning still needs work."*
>
> *"We definitely have a criteria. Scrum kind of forces you to prioritize. We don't necessarily do the right choices every time, but we make decisions anyway because it's impossible not to."*
>
> *"The major change is the fact that we make clear decisions about what to do and what to leave out during the sprint planning sessions. Even if we still didn't have better criteria the prioritization is better defined and everybody is present while doing it."*

It seems like the time-pacing idea behind Scrum sprints is working well in the opinion of both the developers and managers. Also the fact that all stakeholders are present during the sprint planning session makes it clearer for everybody to understand what has been decided about priorities.

Statement 16, "*All the ongoing and immediately upcoming development activities that require attention from the developers are managed as an explicit portfolio*" also had a median answer of 4 during the first round. Especially the developers considered it a problem, since developer median was 4.5. During the second round the overall median has decreased to 2 and also the developer median to 3. Here are some interview quotes to explain the difference:

> *"At least we are better knowledged about the activities. It is also easier to refer to an activity now when we are able to speak about a story or a task."*

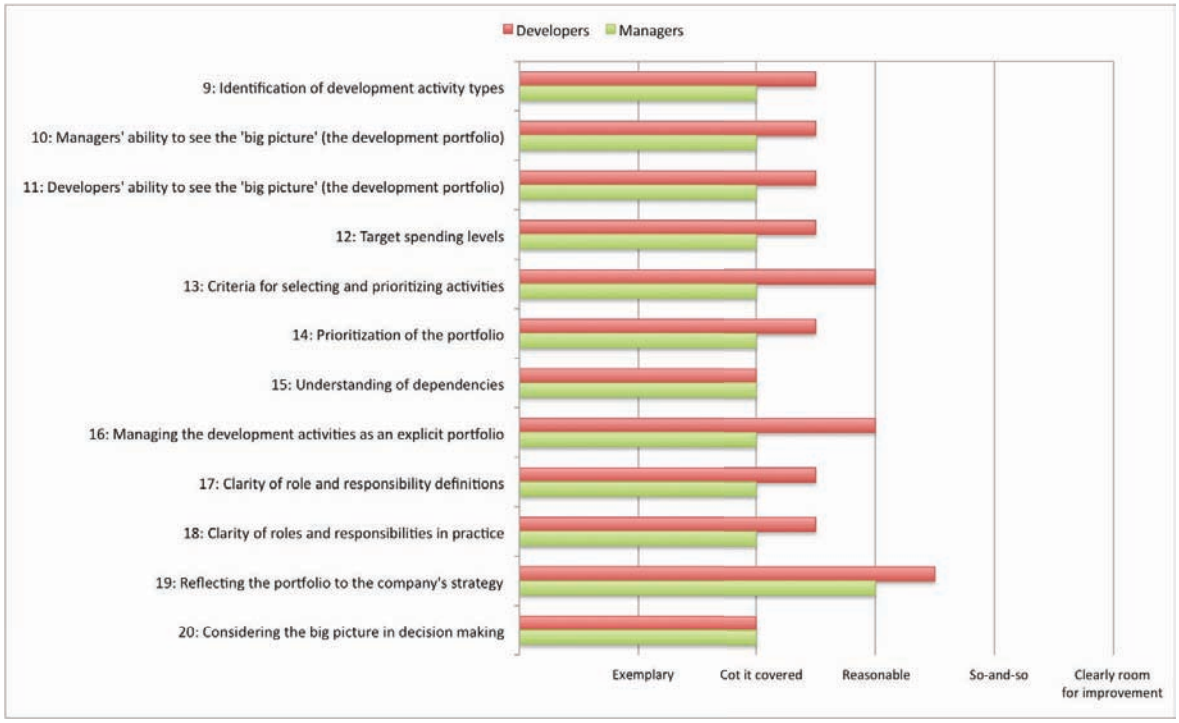


Figure 20: Lifestyle issues in health barometer round 2, developers vs. management

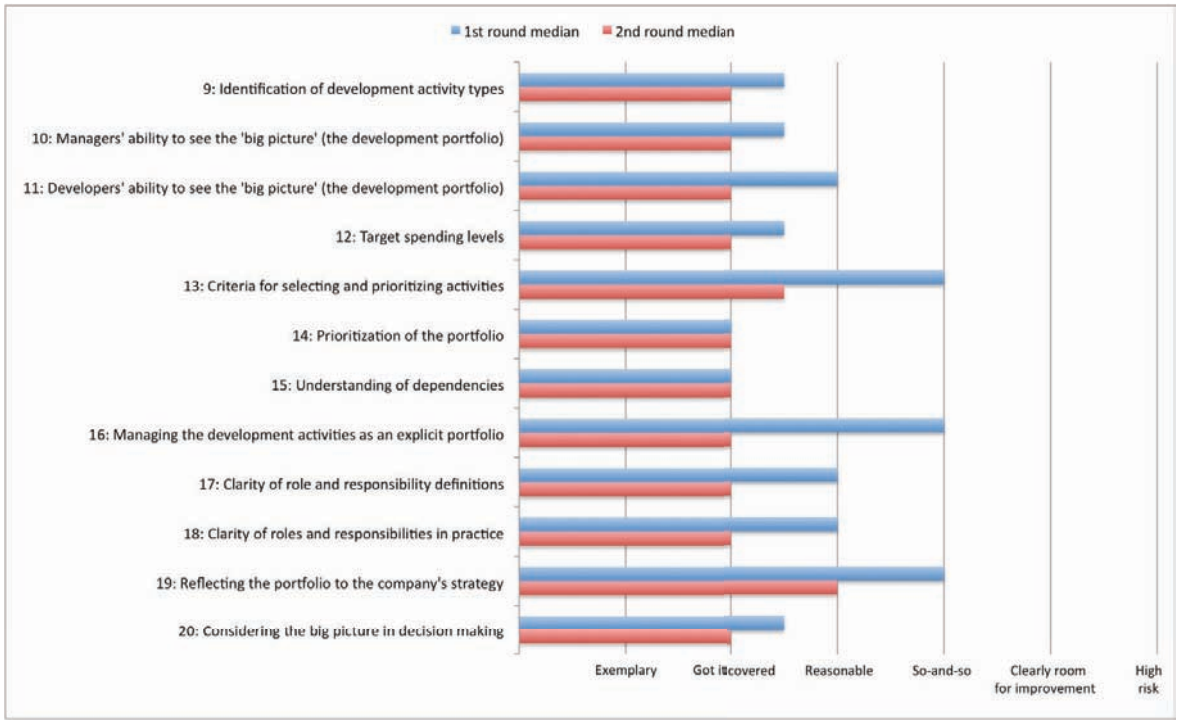


Figure 21: Comparison of lifestyle issues between rounds 1 and 2

> *"Some things still go a bit around the process and when we have some kind of crisis things get difficult. But otherwise yes, they are managed as a whole."*

> *"I suspect that the improvement has happened because we reserve time for individual activities and by that way are able to keep track of the situation as a whole. Because we do proper effort estimation. Now when a developer has something under work, no-one thinks about disturbing because they can see from Agilefant that there is something under construction."*

Statement 17, *"We have defined who are responsible for development portfolio management"*, had the largest interquartile range during the first round. In the second round the median answer was 2, but more importantly the interquartile range fell from 2,5 to 0,5. This was mostly due to the fact that the developer median decreased from 4 to 2,5.

Statement 19, "*We actively reflect the content of the development portfolio to the company's strategy*", had the largest median answer of the lifestyle issues during the second round. The median decreased from 4 to 3, but the topic still needs improvement. The interview quotes included the following:

> *"It is not possible to throw in new work on a tight schedule anymore, which means that the stories must be considered in time. It feels like this makes the strategy more visible."*

> *"The strategy is such an indistinct thing that it feels like it is reflected but I still can't understand it. It is maybe a bit more clear now, but still quite a mystery."*

> *"It helps that some kind of reflection is done formally at least once in three weeks during the iteration change day. It has improved this."*

All in all it can be stated, that just like with the hereditary factors, none of the lifestyle issues received a highed median answer on the second round. The developers were still clearly more pessimistic about the situation, but the overall trend was positive.

6.2.3 Symptoms

Figures 22 and 23 show the results of the symptoms section of the health barometer round 2. Figure 22 shows the median answers of managers and developers compared with each other and figure 23 the median answers of the two rounds compared.

Statements 23, 24, 25 and 35, which deal with resource alignment were all found problematic during the first round and had a median answer of 5. At the second round the statement 23, "*In addition to my main responsibility, I also have other, time-demanding responsibilities*" still had the same median answer, but statements 24, "*We have too many*

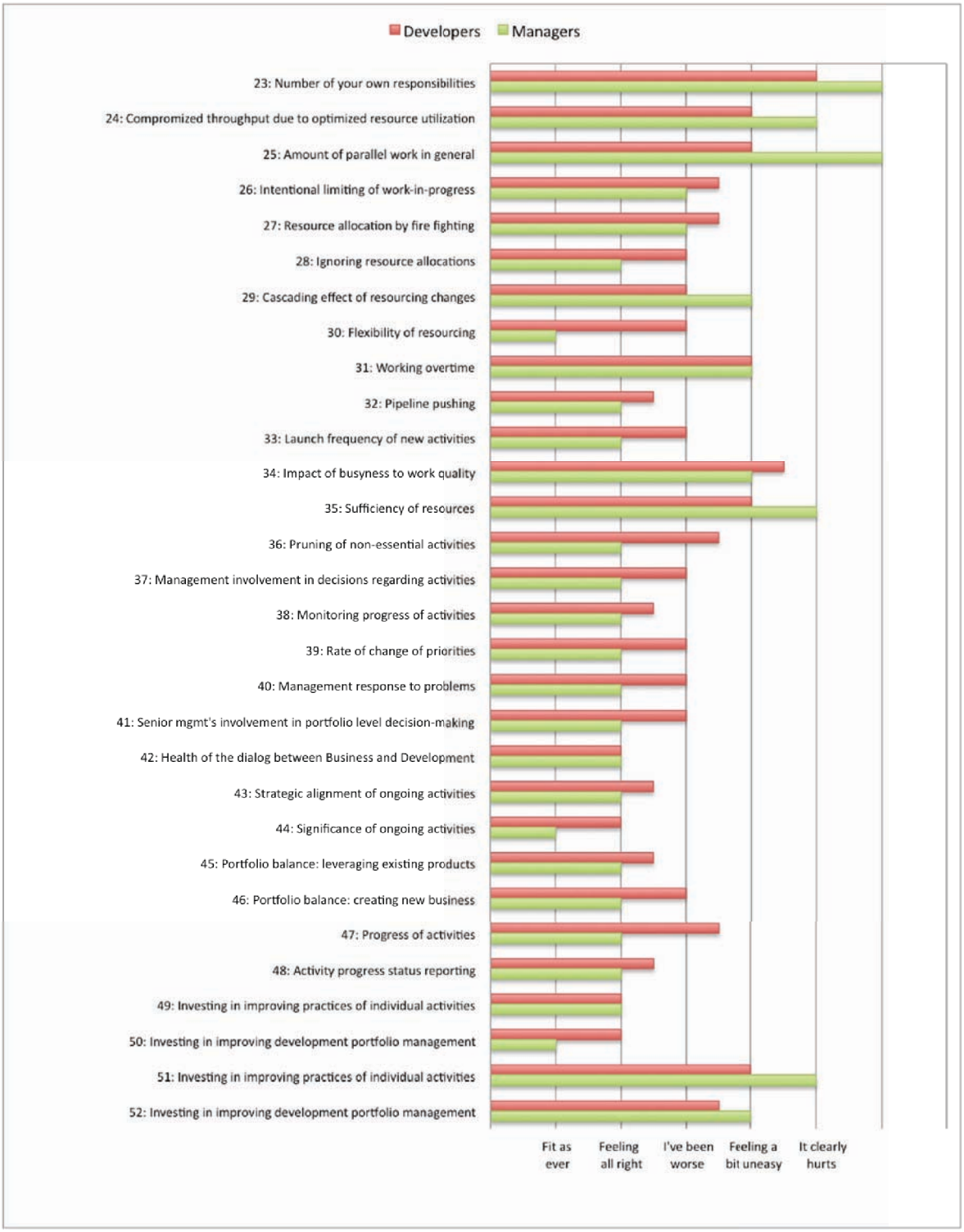


Figure 22: Symptoms in health barometer round 2, developers vs. management

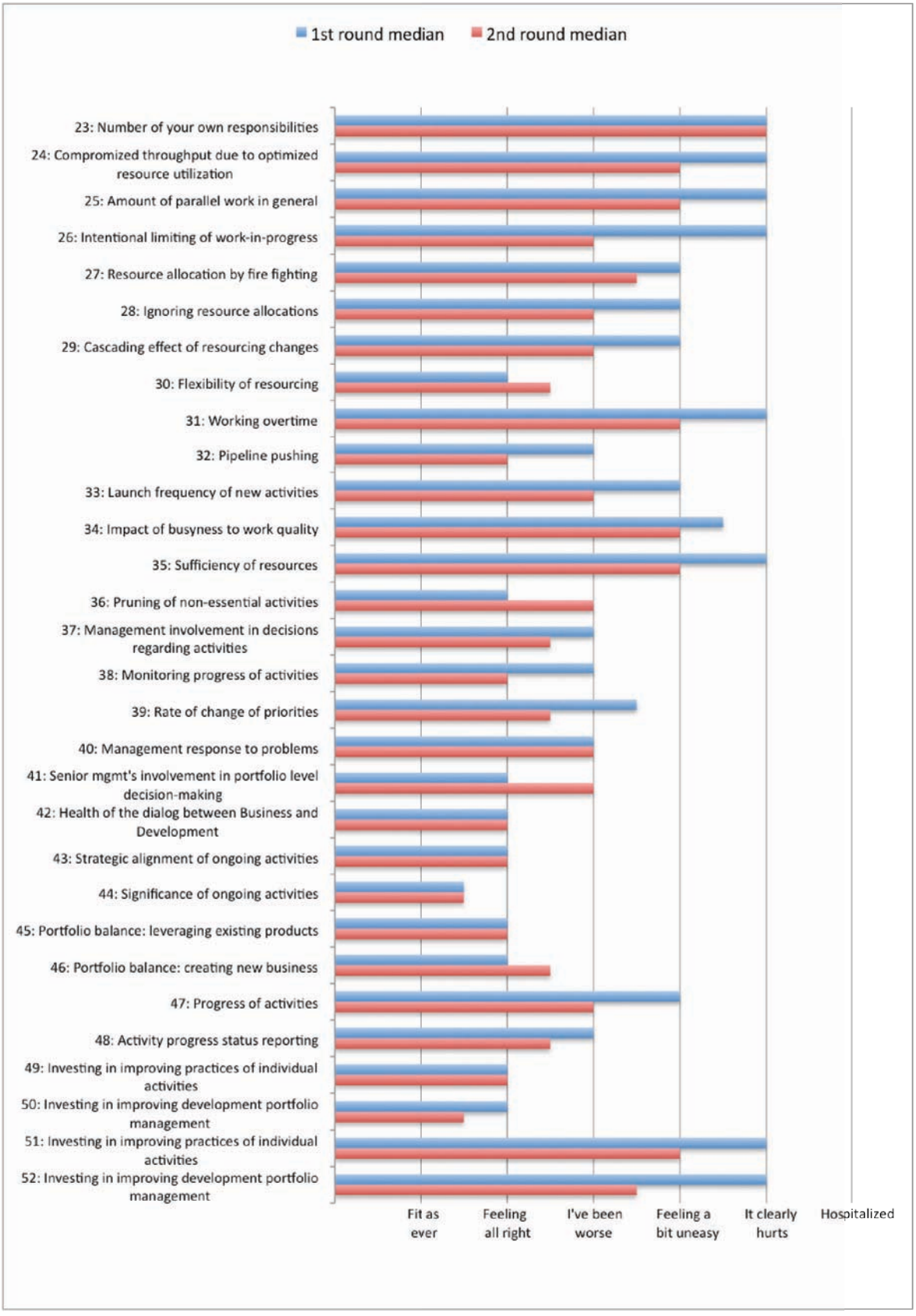


Figure 23: Comparison of symptoms between rounds 1 and 2

parallel ongoing activities", 25, "*A single person is usually assigned to only one activity at the same time*", and 35, "*We have enough resources in proportion to the amount of work*", had now a median answer of 4. Because no new staff was hired during the observation period and the work responsibilities of the development team stayed similar, it can be assumed that the new process had a positive impact on the perceived busyness of work. The following presents some interview quotes concerning these statements.

Statement 23:

> *"It's just like before, and still feels like there are too many of them."*
>
> *"I don't know if it has really changed. We plan things better beforehand, but the same responsibilities are still there."*

Statement 24:

> *"This has improved a little."*
>
> *"The worst pressure has relieved. The most pleasant thing about the current situation is that the managers don't come asking about the status of a task in the last minute. The biggest reason for this is the fact that both managers and developers now share a vision about what is actually under work at the moment."*

Statement 25:

> *"I think that in total this has improved. The key word here is "parallel", I mean that even if someone has many projects going on at the same time, it is now possible to consentrate on them one at a time."*
>
> *"In my opinion this has improved at the development side, but elsewhere no. This is just my opinion, the developers are the ones who really know."*

Statement 35:

> *"The situation has improved a little. However, the situation is still not really good."*
>
> *"It feels like there are more resources available now, but the situation is not too positive anyway."*

Statement 34, "*Our employees have too much work to do and quality of work suffers from it*", had a management median of 4 on both rounds, but the developer median dropped from 5,5 on the first round to 4,5 on the second round. However, the median answer was still 4,5 on the second round so this issue can be considered problematic. The interviews from the second round did not reveal any clear reasons for the improvement.

Statement number 39, "*The priority ranking of activities changes constantly*", had some disagreement during the first round. At the second round the interquartile range was still the same 1,25 as during the first round, but the overall median decreased from 3 to 2. The interview quotes included the following:

> *"It has improved. Thanks to Scrum the ranking doesn't really change except when something accute happens or if new information concerning a story comes up. The ranking is inspected mostly during sprint planning like it should."*
>
> *"The priority ranking doesn't really change, but the order in which things are done does."*
>
> *"It has become visible for everyone, but it still does happen quite a lot."*

The statement 47, "*Ongoing activities are behind schedule*", was one of the most critical during the first round, as it had a developer median of 6 and an overall median of 4. At the second round the overall median declined to 2 and the developer median to 3,5, so a clear improvement was made. Here are some interview quotes:

> *"In my opinion we have been able to learn quickly what we can accomplish in a single sprint. It's a really important skill to understand our own pace, were it fast or slow. This way we know in advance what we can promise."*
>
> *"We have been catching up a little. At least all the stuff using Scrum have been pretty well on schedule. Maybe the timetables are also more realistic now."*
>
> *"Stuff isn't always on time, but at least we have the schedule now. It is also a big change, that because of the iterations the schedule is now clear to everyone."*

Statements 30, 36, 41 and 46 had a larger median answer at the second round compared to the first. It seems natural that statement 30, "*Resource commitments are too rigid for leveraging suddenly emerging opportunities*", is agreed more after switching from an ad-hoc process to a more rigidly defined one.

However, the statement 36, *"Activities are never killed"*, is something that could be assumed to improve after starting to use Scrum. The increase in the median aswer is only 0,5, and it is caused by only two of the seven respondents changing their answers between the rounds, but the finding is still interesting. The interview quotes include the following:

> *"We sometimes understand client accounts as projects, and this way the projects only end if the client relationship is terminated. So this returns to*

> *the fact that we haven't clearly defined the consept of project. Maybe I should have answered: "Don't know""*

Like the quote suggests, it is possible that participants had different definitions for a project and that this might have caused disagreement in the answers. Anyhow, this topic needs to be considered while doing further process improvement actions in the case company.

Statement 41, which says: "*Senior management is actively involved in portfolio-level decision making*", had a median answer of 2 at the first round but median of 3 during the second. The CEO of the company changed after the first sprint of the observation period, which could have caused the change in answers at least partly. The new CEO was clearly more involved in the sales direction, whereas the old focused more on the development. Some of the interview answers supported this interpretation:

> *"This has changed a bit, or if we only consider the senior management to be the CEO it has changed quite a lot."*
>
> *"We have a new CEO and it is not necessary for her to participate in everything. My answer does not indicate what would be the preferred level in my opinion."*
>
> *"Previously the involvement was even a bit uncontrollable, now we could use a bit more."*

The statement 46, *"We have a sufficient amount of product or service development projects that aim for new business"*, also received a larger median answer during the second round. The median increased from 2 to 2,5. This was partly a conscious decision due to some reliability problems experienced with the existing services during the time. It was decided to use resources in stabilizing the services in order to free workforce for new product development later during the year 2012.

6.2.4 Overview

As a whole the second health barometer round showed a clear positive trend in the answers. The medians of the median answers of each of the three categories from both of the rounds are shown in table 9, and from it we can see that improvement was made in the lifestyle issues and symptoms. Also the overall median of median answers declined by 0,5 units.

The hereditary factors did not experience a lot of change, but this was expected since the observation period between the portfolio managemen health barometer rounds was only three months in duration. However, even in the hereditary factors a positive trend could be observed since all of the three issues found most critical during the first health barometer round had a smaller median answer during the second round.

The largest difference between rounds was found in the lifestyle issues, which seems logical since these topics are the ones mostly targeted by the Scrum process. All of the statements in this category received a similar or smaller median answer on the second round. It should be noticed that the difference between managers and developers was present in both rounds, the developers being more pessimistic.

The statements concerning symptoms also experienced an improvement during the second round, but some issues were also considered having suffered from the process improvements. This can be affected by other actions - such as the CEO change or critical situations in some projects - besides the new process, but these issues have to be kept in mind when doing future process improvement work.

Table 9: Overall median answer comparison

Section	Round 1 median	Round 2 median
Hereditary factors	3	3
Lifestyle issues	2,75	2
Symptoms	3,25	3
Overall	3	2,5

7 CONCLUSIONS AND DISCUSSION

This section presents the conclusions that can be drawn from the results presented in the previous section. Also the validity of the results, future process improvement guidelines for the case company and future research possibilities are discussed.

7.1 Conclusions

The conclusions of this study are presented by providing answer suggestions for the research questions and by suggesting a solution to the research problem. The first two research questions are answered based on the literature study presented in section 2. The other research questions are answered based on the research done in this thesis.

1. Why can it be assumed that improving development portfolio management could help solving the problems experienced by the case company?

The problems that were recognized by the employees of case company before the start of this study were according to existing literature all found to be typical symptoms of inadequate portfolio management. The majority of problems recognized in the case company revolve around resource alignment and hence resource alignment is also the main outcome of portfolio management activity this result is reasonable.

Based on these findings it is logical to assume that improving development portfolio management could help solving the problems experienced by the case company.

2. Why can it be assumed that agile software development process could help solving the problems experienced by the case company?

The literature was not found to prove agile methods to be always superior when compared to more traditional software development methods, but several indicators were found that suggested that the case company would benefit from using agile methods. The findings included:

- Agile methods are reported to be easy to adopt
- Agile methods seem to work well in different kinds of environments
- They improve the ability to focus on current work
- Practitioners are relying heavily on agile methods
- The concept of time pacing helps manage the busyness and workload of developers

Furthermore, it can be stated that no evidence was found suggesting that agile methods would not work for the case company. Since all found evidence indicates that agile

methods would be beneficial for the case company it seems reasonable to assume that adopting one would help solving the problems recognized.

Table 10: Results of statements conserning research question 3

Statement	1st round		2nd round	
	Median	Interquartile range	Median	Interquartile range
9. We have identified the different types of activities development people spend their time on	2.5	1	2	0.5
12. I understand how much time, from a business perspective, I should spend on different types of activities	3	1	2	1
13. We have criteria for prioritizing our ongoing development activities	4	2	2.5	1.25
14. I understand the priorities between ongoing activities	2	1.25	2	1
15. I understand the dependencies of the ongoing activities	2	1	2	0.25
19. We actively reflect the content of the development portfolio to the company's strategy	4	2	3	1.5
29. Changes in resourcing for one activity cause uncontrolled changes in other activities	4	1	3	1.25
32. When planning product releases or marketing offers, we plan how to resource the work in practice	3	2	2	0.25
Total average	3.06	1.41	2.31	0.88
Total median	3	1.125	2	1

3. ***Does the ability of the different stakeholders to make informed decisions about resource alignment improve after implementing the process model?***

The statements concerning this research question together with the median answers and interquartile ranges from both health barometer rounds are presented in table 10. The table also includes medians and averages of the median answers and interquartile ranges for overall comparison. As the table shows, the overall impression of the employees was that the situation improved after starting to use Scrum. Also the interquatile ranges were smaller, which implies that there is a better consensus about the situation. Analysis of the sprint metrics presented in section 7.1 also suggests, that the stakeholders' ability to estimate needed resources improved during the use of the new process.

This evidence suggests that the answer to research question 3 is positive i.e. the ability of different stakeholders to make efficient decisions concerning resource allocation did in fact improve after implementing the suggested process model.

4. Does the ability of the development team to focus on current work improve after implementing the process model?

The statements concerning this research question together with the median answers and interquartile ranges from both health barometer rounds are presented in table 11. As the table shows, the situation with the ability to focus on current work was far from optimal before the implementation of the new process since the average median was above 4. According to the health barometer the situation has improved during the observation period but is still something the case company needs to work on. Additionally, the interquartile range of the answers has increased during the second round, which implies that there is more disagreement about the situation than before starting to use Scrum.

These results can be summarized so, that the ability of the development team to focus on current work seems to have improved for most employees after implementing the new process model, but based on the increased interquartile range some developers have not experienced such an improvement. This is somewhat surprising, since one of the key benefits of Scrum should be the ability to finish things one by one.

However, it is possible that the observation period of three months used in this study was too short for the employees to learn the needed skills to make improvements also in this category. In the first iterations it might feel like the activities needed for running the process were at times interrupting normal work.

Table 11: Results of statements concerning research question 4

Statement	1st round		2nd round	
	Median	Interquartile range	Median	Interquartile range
24. We have too many parallel ongoing activities	5	1.25	4	1.25
25. A single person is usually assigned to only one activity at the same time	5	1	4	2.25
26. We complete one thing at a time and don't shift our attention from one incomplete task to another	5	1.25	3	2
27. Fire fighting describes our work in practice	4	0.5	3.5	1
28. Resources are being shifted from one activity to another regardless of previously agreed assignments	4	1	3	1.25
33. New activities are launched too often	4	0.25	3	1
34. Our employees have too much to do and quality of work suffers from it	4.5	1.25	4	1
39. The priority ranking of activities changes constantly	3.5	1.25	2.5	1.25
Total average	4.38	0.97	3.38	1.38
Total median	4.25	1.125	3.25	1.25

5. Does the ability of the stakeholders to follow how their decisions are reflected into the daily work of developers improve after implementing the process model?

The statements concerning this research question together with the median answers and interquartile ranges from both health barometer rounds are presented in table 12. The overall median answer has decresed by approximately one unit and the interquartile ranges also have a slight decrease. These results suggest that the ability of stakeholders to follow how their decisions are reflected in the daily work has improved.

Table 12: Results of statements concerning research question 5

Statement	1st round		2nd round	
	Median	Interquartile range	Median	Interquartile range
10. Business people are able to see the 'big picture' of ongoing activities	2.5	1.25	2	0.5
11. Development people are able to see the 'big picture' of ongoing activities	3	0.25	2	1.25
16. All the ongoing and immediately upcoming activities that require attention from the developers are managed as an explicit portfolio	4	0.5	2	1
38. The real status of activities is known in development portfolio –level decision making	3	0.75	2	0.25
42. The dialogue between business and development people works	2	1	2	0
48. Progress of ongoing activities is reported optimistically	3	1	2.5	1
Total average	2.92	0.79	2.08	0.67
Total median	3	0.875	2	0.75

The main research problem stated in section 1.3 was:

> ***Can adopting an agile software process alleviate the problems recognized in the daily development activities of the case company?***

Based on the results presented in previous chapters and the presented answers to the research questions it is possible to further conclude that this study suggests a positive answer to the research problem. There seems to be no clear evidence, neither in literature nor in this study, against agile software process adoption being able to improve the situation of companies having similar issues as the case company. The results obtained in this study suggest with very little exceptions that both developers and managers perceived the everyday development activity of the case company to be running smoother after starting to use Scrum.

7.2 Validity and limitations of the results

Some of main limitations for the results obtained in this thesis arise from the case study – nature of the research. Applying similar process improvement efforts for other organizations might in practice provide completely different results. Furthermore, since no practical comparison between different process models was conducted, this study can be criticized by stating that any other process model could have worked just as well or

possibly even better than Scrum. However, it is also reasonable to expect that other organizations similar to the case company both in size and nature of problems would benefit from Scrum in the same way.

Comparing the new agile process to the old ad hoc –type of process is difficult since no metrics are available before the adoption of Scrum. The employees might feel like the development is more efficient or the quality is better just because they assume it to be since an effort has been made to improve the process.

7.3 Future process improvement guidelines

Although this study suggests that the case companys development portfolio management improved by implementing Scrum process it must be kept in mind that Scrum alone is no portfolio management tool. In fact, Scrum says very little about how to actually do the portfolio management work and how to connect it with the strategy of the company. It would be a logical extension to the process improvement efforts described in this study to focus more on the portfolio management issues in the future.

Currently the case company is able to plan and execute iterations relatively well and the next step would be to focus more on release- or project-level planning. Also the issue of connecting the development projects and the resource allocation balance with the strategy of the company should be on the agenda when making process improvement actions in the future.

7.4 Future work

The existing literature for agile software development methods and process models offers quite limited insights into selecting the most suitable method from the numerous possibilities. There are case studies and lessons learned –types of articles but they mostly focus on questions like “How to implement a certain agile process?” or “What benefits did a certain agile process have?”. The practitioners are in a need for methods to assess and compare different alternatives.

The literature concerning Scrum often resolves the complex issue of portfolio management by simply stating that the product owner is responsible for managing the different backlogs. Advice on how to do this in practice is quite scarce and also typically presented in lessons learned –types of studies. More scientific proof on how to actually implement the role of product owner in practice and how to perform the numerous tasks attached to the role is needed.

As for the development portfolio management health barometer method, it would be interesting to have some sort of benchmark information on how certain types of hereditary issues affect the organization’s ability to achieve a desired level on lifestyle issues or

symptoms. Is it even possible to have a rigorous enough lifestyle that you don't have to suffer from symptoms if all hereditary issues are against you?

8 REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. 2002, "Agile software development methods – Review and analysis", VTT Publications 478, Espoo, Finland.

Beck, K., et al. 2001, "The Agile Manifesto", Available at: http://www.agilemanifesto.org (accessed February 15th 2012).

Beck, K. 2004, "Extreme Programming Explained: Embrace Chage", second ed., Addison-Wesley, ISBN 978-0321278654.

Benefield, G. 2008, "Rolling out agile in a large enterprise". In Hawaii International Conference on System Sciences, Proceedings of the 41st Annual (pp. 461-461). IEEE.

Boehm, B. 2002, "Get ready for agile methods, with care". IEEE Computer 35 (1) (2002) 64–69.

Chow, T., and Cao, D.-B. 2008. "A Survey Study of Critical Success Factors in Agile Software Projects," The Journal of Systems and Software (81:6), pp. 961-971, Elsevier.

Cockburn, A. 2004, "Crystal Clear: A Human-Powered Methodology for Small Teams", Addison-Wesley, ISBN 0-201-69947-8.

Cooper, R., Edgett, S. & Kleinschmidt, E. 2001, "Portfolio Management for New Products". Perseus Books, 2001.

Cooper, R., Edgett, S. & Kleinschmidt, E. 2002, "Portfolio management: fundamental to new product success," in The PDMA Toolbook for New Product Development, P. Belliveau, A. Griffin, & S. Somermeyer, eds., John Wiley & Sons, Inc., New York, pp. 331-364.

Craddock, A., Fazackerley, B., Messenger, S., Roberts, B. and Stapleton, J. 2008, "DSDM Atern Handbook V2". DSDM Consortium, Kent, UK, ISBN 0-9544832-2-7. Available at: http://www.dsdm.org/atern-handbook/flash.html (accessed December 7th 2012).

Dybå, T. 2005. "An Empirical Investigation of the Key Factors for Success in Software Process Improvement". IEEE Transactions on Software Development, vol 31, no. 5.

Dybå, T. and Dingsøyr, T., 2008. "Empirical studies of agile software development: A systematic review". Information & Software Technology 50(9–10), 833–859.

Heikkilä, V. & Rautiainen, K. 2010, "Chapter 5: Performing a Portfolio Management Health Barometer Study" in Towards Agile Product and Portfolio Management, eds. V. Heikkilä, K. Rautiainen & J. Vähäniitty, Espoo: Aalto University, pp. 72-85

Holford, W. D. and Ebrahimi, M. 2007, "Honda: Approach to Innovation in Aerospace and Automotive/Pick-Up Truck Development: A Dialectical Yet Coherent Firm," in 40th Annual Hawaii International Conference on System Sciences (HICSS-40), Big Island, Hawaii.

Humphrey, W. S. 1989, "Managing the software process", Addison-Wesley.

Jiang, L. and Eberlein, A. 2009, "An Analysis of the History of Classical Software Development and Agile Development". Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA.

Larman, C. 2004. "Agile and iterative development: a manager's guide". Addison-Wesley Professional.

Leffingwell, D. and Behrens, P. 2009, "A User Story Primer", Leffingwell LLC.

Liker, J. K. 2004, "The Toyota way: 14 Management Principles from the World's Greatest Manufacturer". New York: McGraw-Hill.

Mar, K., & Schwaber, K. 2002, "Scrum with XP", referenced 16th December 2012, available at: http://www.informit.com/articles/article.aspx?p=26057

McGrath, M. 1996, "Setting the PACE in product development".

Niazi, M., Wilson, D. and Zowghi, D. 2003. "A model for the implementation of a software process improvement: a pilot study". Proceedings of the Third International Conference on Quality Software, IEEE.

Palmer, S.R., Felsing, J.M. 2002, "A Practical Guide to Feature-driven Development", Prentice Hall, Upper Saddle River, NJ, ISBN 0- 13-067615-2.

Pikkarainen, M., Salo, O., and Still, J. 2005, "Deploying agile practices in organizations: a case study". In Proceedings of the 12th European conference on Software Process Improvement (pp. 16-27). Springer-Verlag.

Poppendieck, M. and Poppendieck, T. 2003. "Lean Software Development – An Agile Toolkit for Software Development Managers", Addison-Wesley, Boston, ISBN 0-321-15078-3.

Poppendieck, M. 2005, "A History of Lean: From Manufacturing to Software Development," in JAOO Conference, Aarhus, Denmark.

Poppendieck, M. and Poppendieck, T. 2010, "Leading Lean Software Development: Results are Not the Point". Boston, MA, USA: Addison-Wesley

Rothman, J. 2009, "Manage your project portfolio: Increase your capacity and finish more projects", Raleigh, NC: Pragmatic Bookshelf.

Royce, W.W. 1970. "Managing the development of large software systems." proceedings of IEEE WESCON. Vol. 26. No. 8. 1970.

Schwaber, K. 1995, "SCRUM Development Process", Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA, 1995, pp117-134)

Schwaber, K. and Beedle, M. 2001, "Agile Software Development with Scrum", Prentice Hall, Upper Saddle River.

Schwaber, K. 2004, "Agile project management with Scrum", Microsoft Press, Richmond WA. ISBN 0-7356-1993-X., ISBN 978-0-7356-1993-7.

Schwaber, K. and Sutherland, J. 2011, "Scrum Guide", October 2011 Release, referenced April 6th 2012. Available at: http://www.scrum.org/Scrum-Guides

Stapleton, J. 2003, "DSDM: Business Focused Development", second ed., Pearson Education, ISBN 978-0321112248.

Vähäniitty, J. 2006, "Do Small Software Companies Need Portfolio Management, Too?" Licenciate Thesis, Helsinki University of Technology.

Vähäniitty, J. 2010, "Chapter 2: Agile Product and Portfolio Management – Crucial for Competitiveness" in Towards Agile Product and Portfolio Management, eds. V. Heikkilä, K. Rautiainen & J. Vähäniitty, Espoo: Aalto University, pp. 31-37.

Vahaniitty, J., Rautiainen, K., Lassenius, C. 2010, "Small software organizations need explicit project portfolio management", IBM Journal of Research and Development, vol. 54, no. 2, pp. 1:1-1:12.

West, D. & Grant, T. 2010, "Agile Development: Mainstream adoption has changed agility", Forrester Research.

Wheelwright, S. & Clark, K. 1992, "Revolutionizing Product Development". New York: The Free Press.

Wikimedia Commons, 2008. Scrum process.svg –image. Updated February 9th 2008, referenced December 6th 2012. Available at: http://en.wikipedia.org/wiki/File:Scrum_process.svg

APPENDIX A: INSTRUCTIONS FOR THE SURVEY

Hei kaikki!

Tässä tulee ohjeet diplomityöprojektiini liittyvään kyselyyn vastaamisesta. Kiitokset vastaajille jo etukäteen!

Healt Barometer (HB) on Aalto-Yliopiston Ohjelmistoliiketoiminnan ja -tuotannon laitoksen ATMAN-tutkimusprojektissa kehitetty menetelmä yrityksen tekemissalkun hallintaan liittyvien prosessien ja menettelytapojen arviointiin. Termit "tekemissalkku" ja "tekemissalkun hallinta" on määritelty tarkemmin alla. HB:n avulla voidaan kartoittaa kehitystarpeita yrityksen toiminnassa, jotta parannustoimet osataan kohdistaa oikein. Tavoitteena on siis helpottaa kaikkien työntekoa!

Tämä viesti sisältää ohjeet kyselyyn vastaamiseen. Lähetän kaikille henkilökohtaisesti erillisessä viestissä vielä tunnukset, joilla kyselyyn pääsee kirjautumaan vastaamista varten. Deadline vastaamiselle on 11.11., mutta toivoisin että vastaatte ennen sovittua haastatteluaikaa, jotta haastattelu voidaan toteuttaa vastausten pohjalta. Kyselyyn vastaamiseen menee ensimmäisellä kerralla noin 30 minuuttia. Lue nämä ohjeet ennen vastamista!

Kyselyyn vastataan seuraavasti:

1) Siirry osoitteeseen http://example.com/survey

2) Kirjaudu sisään käyttäjänimellä ja salasanalla, jonka saat hetken kuluttua erillisessä sähköpostiviestissä

3) Siirry kyselyyn valitsemalla avoinna oleva kierros (Round 1 tässä tapauksessa)

4) Aloita valitsemalla vastuualueesi yrityksessä. Valitse parhaiten toimenkuvaasi vastaava vastuualue.

5) Täytä kysely loppuun valitsemalla kuinka samaa mieltä olet esitettyjen väittämien kanssa. Valitse vastauksesi omasta näkökulmastasi, älä pyri antamaan koko yrityksen "keskiarvovastausta". Vastaa realistisesti ja vältä sensuuria, näin tuloksetkin ovat parempia! Jos olet epävarma siitä, ymmärsitkö väittämän oiken, merkitse kysymys tai huomio kommenttikenttään. Jos ymmärsit väittämän, mutta et tiedä vastausta, valitse "En tiedä". Vastauksesi eivät päädy muiden kuin Juhon luettaviksi. Tulokset esitetään siten, että niistä ei voida yksilöidä kenenkään henkilökohtaisia vastauksia.

6) Voit tallentaa vastauksesi valitsemalla kyselyn alalaidasta "Save". Vastauksia ei tarvitse antaa kaikkia yhdellä kertaa, vaan voit tarvittaessa keskeyttää vastaamisen ja tallentaa tilanteen palataksesi siihen myöhemmin. Järjestelmässä on sisäänkirjautumisen

aikakatkaisu, eli muista tallentaa jos keskeytät vastaamisen pidemmäksi aikaa, jotta vastauksesi eivät häviä!

Tässä muutama määritelmä suomeksi ja englanniksi kyselyn kontekstin ymmärtämisen tueksi:

TEKEMISSALKKU viittaa kaikkien "kehitysporukan" (eli teknisen ja/tai tuotekehityksen henkilöstön) huomiota vaativien, meneillään sekä välittömästi suunnitteilla olevien "tekemisten" kokonaisuuteen. Esimerkkejä tyypillisistä tekemisten tyypeistä ovat tuotekehitysprojektit, ylläpito, asiakaskohtainen kehitys, toimitukset, asiakaspalvelu, koulutus, konsultointi ja myynnin tuki. Tekemisen tyypeillä ei kuitenkaan tarkoiteta perinteistä ohjelmistokehityksen jakoa määrittelyyn, suunnitteluun, koodaukseen ja testaukseen.

TEKEMISSALKUN HALLINTA on tekemissalkun ajan tasalla pitämisestä vastaava päätöksentekoprosessi. Tekemissalkun hallinnassa priorisoidaan tekemisiä (esim. tuotekehitysprojektit) ja päätetään niiden resursoinnista. Tekemissalkun hallinnassa päätetään myös miten äkillisesti ilmaantuvat tekemisten väliset konfliktitilanteet hoidetaan.

IN ENGLISH:

THE DEVELOPMENT PORTFOLIO is the set of ongoing and upcoming activities that require attention from the "development people" (e.g. product development and/or technical resources). Common types of development activity types are e.g. release-based product development projects, customer-specific development, maintenance, deliveries, customer service, training, consultation, sales support, etc. However, specification, design, coding and testing are NOT types of activities we are looking for here.

DEVELOPMENT PORTFOLIO MANAGEMENT is the decision process for updating and revising the development portfolio. In development portfolio management, development activities (e.g. projects) are prioritized and resourced. Development portfolio management is also responsible for appropriately resourcing the handling of suddenly emerging urgencies.

Jos kyselyn tai sen vastaamisen kanssa tulee ongelmia (teknisiä tai muita) ota yhteys Juhoon (juho@email.fi / 040 1234567).

Kun sekä haastattelut että kyselyt on tehty ja tulokset on analysoitu pidetään kaikille yhteinen tilaisuus, jossa tulokset käydään läpi. Palaan asiaan tarkemmin, kun tarkka aika ja paikka on saatu sovittua.

APPENDIX B: SURVEY ANSWERS

	Round 1					Round 2					Round 1								Round 2						
Issue	25% Quartile	75% Quartile	Median	Developer median	Management median	25% Quartile	75% Quartile	Median	Developer median	Management median	D1	D2	D3	D4	M1	M2	M3	C1	D1	D2	D3	D4	M1	M2	M3	C1
1: Leveraging customer-specific activities for product development	5	5	5	5	5	3.75	5	4.5	4	5	5	6	5	2	5	5	5	N/A	5	2	5	3	5	4	5	4
2: Multiple roles and responsibilities	4.75	5	5	5	5	4	5	4	4.5	4	5	5	5	5	4	6	5	3	4	5	4	5	4	4	5	2
3: Dependency on cash flow	2	3.5	3	3	3	3	4	3	3.5	3	2	N/A	4	3	3	5	2	2	2	4	4	3	3	4	3	3
4: Clarity of strategy: Definition	2.75	3.25	3	3.5	2	2.75	3	3	3	3	3	4	4	3	2	3	2	3	3	2	3	3	3	2	6	3
5: Clarity of strategy: Communication	3	3	3	3	3	2.75	3	3	3	3	5	3	3	3	2	3	3	3	3	2	3	3	2	3	6	3
6: Appropriateness of incentive systems	1	2.5	1	3	1	1	2	1	2	1	N/A	3	3	2	1	1	1	1	N/A	1	3	2	1	1	2	1
7: Appropriateness of organisational structure	2	2.25	2	2.5	2	2	2	2	2	2	2	3	3	2	2	2	2	2	2	1	3	2	2	2	2	2
8: Health of individual activities' practices	3	4	3.5	3.5	4	2	3	3	3	2	6	3	4	3	4	3	4	3	3	3	4	2	3	2	2	3
9: Identification of development activity types	2	3	2.5	3	2	1.75	2.25	2	2.5	2	2	3	4	3	2	2	3	2	1	3	3	2	2	2	2	1
10: Managers' ability to see the 'big picture' (the development portfolio)	2	3.25	2.5	3.5	2	1.75	2.25	2	2.5	2	3	4	3	4	2	2	2	2	2	3	2	3	2	1	2	1
11: Developers' ability to see the 'big picture' (the development portfolio)	2.75	3	3	3	3	1.75	3	2	2.5	2	3	3	5	3	3	3	2	2	1	3	3	2	3	2	2	1
12: Target spending levels	2	3	2.5	3	2	2	3	2	2.5	2	4	3	3	3	2	2	2	2	2	3	3	2	2	2	2	3
13: Criteria for selecting and prioritizing activities	2.5	4.5	4	4.5	2	1.75	3	2.5	3	2	5	4	3	5	2	2	4	N/A	1	3	3	3	2	1	4	2
14: Prioritization of the portfolio	2	3.25	2	3.5	2	2	3	2	2.5	2	4	4	3	2	2	2	2	2	3	3	2	2	2	2	3	2
15: Understanding of dependencies	2	3	2	2.5	2	2	2.25	2	2	2	3	3	2	2	3	2	2	2	2	3	2	1	2	2	2	2
16: Managing the development activities as an explicit portfolio	4	4.5	4	4.5	4	2	3	2	3	2	5	4	5	4	2	4	4	N/A	2	3	4	3	2	2	1	2
17: Clarity of role and responsibility definitions	2.5	5	3	4	2	1.75	2.25	2	2.5	2	6	3	5	3	2	2	5	N/A	3	2	3	2	2	2	1	1
18: Clarity of roles and responsibilities in practice	2.75	3	3	3	3	2	3	2	2.5	2	5	3	3	2	2	3	3	3	4	2	3	2	2	3	2	2
19: Reflecting the portfolio to the company's strategy	3	5	4	5	3	3	4.5	3	3.5	3	6	4	5	5	3	2	3	N/A	5	3	3	4	3	2	5	N/A
20: Considering the big picture in decision making	2	3	2.5	3	2	2	2.25	2	2	2	3	3	4	3	1	2	2	2	2	2	2	3	1	2	4	2
21: Number of ongoing activities	6	14	10	6	14	6.25	10.25	7	5.5	13.5	N/A	6	N/A	6	10	14	14	N/A	N/A	4	N/A	7	7	20	>10	A lot
22: Number of activities you are involved in	3.5	5.5	4	4	7	3.5	9	5	4	11	N/A	3	4	4	7	8	4	3	N/A	3	4	5	7	15	11	3
23: Number of your own responsibilities	5	6	5	5	5	5	6	5	5	6	6	4	5	5	5	6	5	6	6	4	5	5	5	6	6	5
24: Compromized throughput due to optimized resource utilization	3.75	5	5	5	5	3.75	5	4	4	5	5	6	3	5	3	5	5	4	4	4	3	5	4	5	5	3
25: Amount of parallel work in general	5	6	5	5.5	5	3.75	6	4	4	6	6	6	4	5	6	5	5	5	6	4	4	3	6	3	6	4
26: Intentional limiting of work-in-progress	3.75	5	5	5	3	2	4	3	3.5	3	5	5	5	5	3	2	5	4	3	4	2	4	3	1	5	2
27: Resource allocation by fire fighting	4	4.5	4	4	4	3	4	3.5	3.5	3	4	6	4	4	4	5	4	N/A	2	4	3	4	3	4	3	4
28: Ignoring resource allocations	3	4	4	4	4	1.75	3	3	3	2	3	4	4	4	2	4	4	3	1	3	3	3	1	3	2	3
29: Cascading effect of resourcing changes	4	5	4	4.5	4	2.75	4	3	3	4	5	5	4	4	4	5	4	3	3	4	3	3	2	4	4	2
30: Flexibility of resourcing	1	3	2	2.5	1	1	3.25	2.5	3	1	1	2	3	3	5	1	1	2	1	4	3	3	4	1	1	2
31: Working overtime	4.75	6	5	6	5	2.5	4.25	4	4	4	6	4	6	6	5	5	5	2	5	3	4	4	4	5	1	1

	Round 1					Round 2					Round 1								Round 2							
Issue	25% Quartile	75% Quartile	Median	Developer median	Management median	25% Quartile	75% Quartile	Median	Developer median	Management median	D1	D2	D3	D4	M1	M2	M3	C1	D1	D2	D3	D4	M1	M2	M3	C1
32: Pipeline pushing	2	4	3	4	2	2	2.25	2	2.5	2	5	4	4	3	2	2	3	2	2	3	3	2	2	1	2	2
33: Launch frequency of new activities	3.75	4	4	4	4	2	3	3	3	2	4	4	4	4	4	3	4	2	3	3	3	3	2	2	3	2
34: Impact of busyness to work quality	4	5.25	4.5	5.5	4	4	5	4	4.5	4	6	6	4	5	4	5	4	3	5	4	4	5	4	4	5	4
35: Sufficiency of resources	4	5	5	4.5	5	4	5	4	4	5	6	5	4	4	5	5	5	3	5	4	4	4	4	5	5	4
36: Pruning of non-essential activities	2	3.5	2	3	2	2	4	3	3.5	2	1	3	N/A	4	2	2	2	5	1	3	4	4	2	2	N/A	5
37: Management involvement in decisions regarding activities	1.75	3	3	3	2	1.75	3.25	2.5	3	2	1	3	3	3	4	1	2	3	1	4	3	3	2	1	2	5
38: Monitoring progress of activities	3	3.75	3	3	3	2	2.25	2	2.5	2	3	N/A	4	3	2	3	5	N/A	2	3	3	2	2	2	2	1
39: Rate of change of priorities	3	4.25	3.5	4.5	3	2	3.25	2.5	3	2	6	5	4	3	3	4	3	2	2	4	3	3	2	4	2	2
40: Management response to problems	2.5	3.5	3	4	3	2	3.5	3	3	2	5	4	N/A	3	2	3	3	2	4	3	N/A	3	2	2	4	2
41: Senior mgmt's involvement in portfolio level decision-making	1.5	2.5	2	3	2	2	3.25	3	3	2	1	3	N/A	4	1	2	2	2	3	4	3	3	1	4	2	2
42: Health of the dialog between Business and Development	2	3	2	3	2	2	2	2	2	2	3	4	3	2	2	1	2	2	2	2	3	2	2	1	2	2
43: Strategic alignment of ongoing activities	1.75	2.25	2	2.5	2	1.75	2.25	2	2.5	2	2	3	2	3	2	1	2	1	2	3	2	3	2	1	2	1
44: Significance of ongoing activities	1	2.25	1.5	2.5	1	1	2	1.5	2	1	1	3	2	3	1	1	2	1	1	3	2	2	1	1	1	2
45: Portfolio balance: leveraging existing products	1.75	2.25	2	2	2	1	3	2	2.5	2	1	2	3	2	4	2	2	1	1	3	3	2	5	2	1	1
46: Portfolio balance: creating new business	1.75	3	2	2.5	2	1.75	3	2.5	3	2	1	2	3	3	3	2	2	1	3	3	3	4	2	2	1	1
47: Progress of activities	4	5.5	4	6	4	2	3.25	3	3.5	2	6	6	N/A	5	3	4	4	4	4	3	3	4	2	3	2	2
48: Activity progress status reporting	3	4	3	4	3	2	3	2.5	2.5	2	4	6	N/A	3	3	3	4	2	3	4	2	2	3	2	1	3
49: Investing in improving practices of individual activities	2	2.25	2	2	2	2	2	2	2	2	2	3	2	2	2	2	3	2	2	3	2	2	2	2	2	1
50: Investing in improving development portfolio management	1.75	2.25	2	2.5	2	1	2	1.5	2	1	1	3	2	3	2	2	1	2	1	2	2	2	2	1	1	1
51: Investing in improving practices of individual activities	4.75	5	5	5	5	4	5	4	4	5	6	4	5	5	5	4	5	5	4	4	4	5	5	4	5	4
52: Investing in improving development portfolio management	4.75	5.25	5	5.5	5	3	4	3.5	3.5	4	6	6	4	5	5	4	5	5	1	3	4	4	4	3	5	3